

数据通道

数据通道

背景要求

基本架构

主要实现技术：io拷贝

流程图（动态通知SA连接）

后续优化

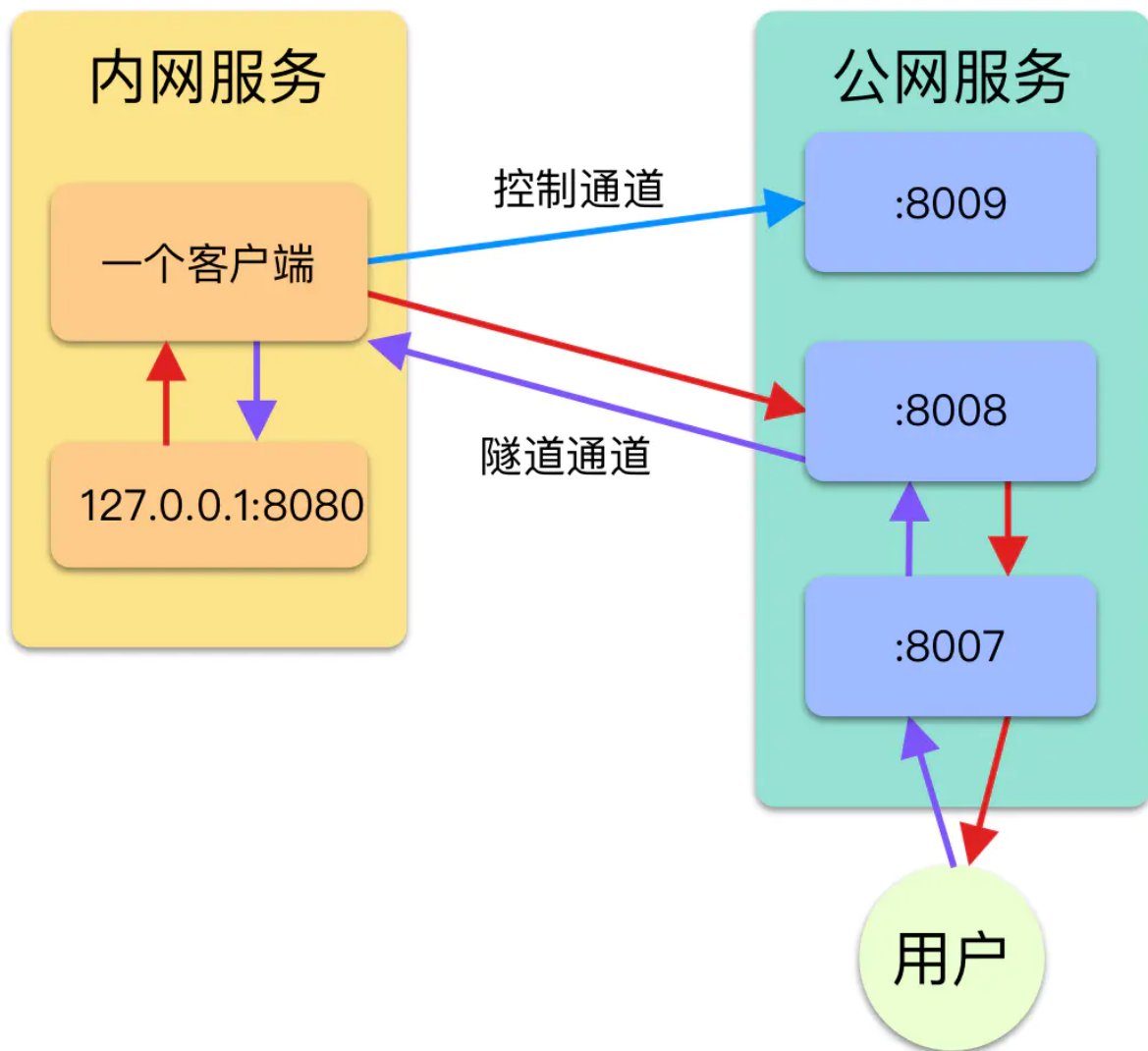
背景要求

1. 支持TCP、HTTP、WS请求
2. SA作为Socket客户端接入，与实际客户端建立通道
3. 支持大数据量，多客户端接入

基本架构

描述：

1. 先建立控制通道。每个用户客户端请求，SC会通过控制通道通知SA请求一个TCP连接建立通道
2. 当SA收到通道端口的数据再次IO拷贝到接受HTTP端口
3. 后续：SA可根据控制通道通知的消息，判断接受到的数据是否需要转发到HTTP端口



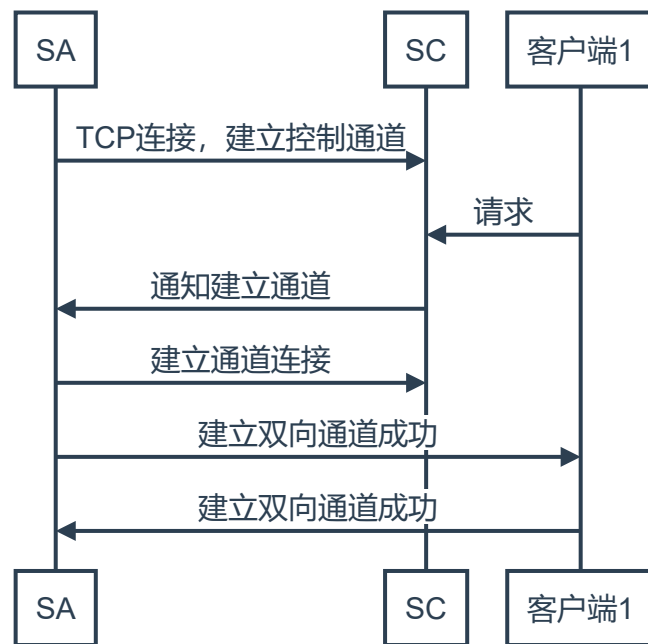
主要实现技术：io拷贝

```
io.Copy(src, dest)
```

```
func Join2Conn(local *net.TCPConn, remote *net.TCPConn) {
    go joinConn(local, remote)
    go joinConn(remote, local)
}

func joinConn(local *net.TCPConn, remote *net.TCPConn) {
    defer local.Close()
    defer remote.Close()
    _, err := io.Copy(local, remote)
    if err != nil {
        log.Println("copy failed ", err.Error())
        return
    }
}
```

流程图（动态通知SA连接）



优点：1. 能动态增加或减少SA连接。

缺点：1. 通知SA连接可能造成延迟

后续优化

1. 优化连接池map管理
2. 做好连接复用