第二届 eBPF开发者大会
www.ebpftravel.com

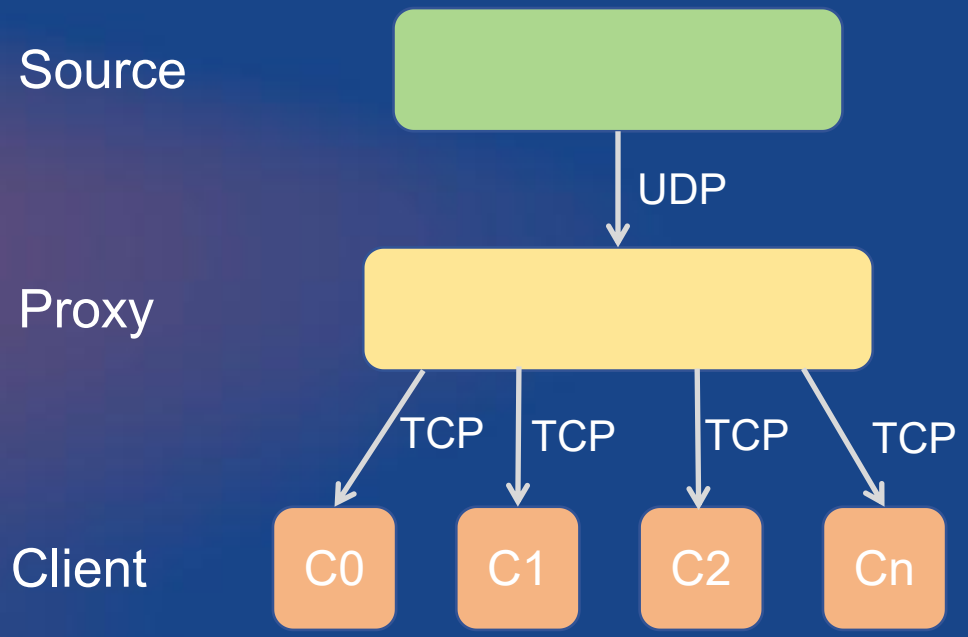
基于ebpf的socket代理转发方案实践

陆　云
麒麟软件有限公司

中国·西安

背景介绍

数据转发模型



用户进程使用send/recv系统调用进行转发

对转发延迟敏感，延迟越小越好

客户端较多的情况下，Proxy转发存在性能瓶颈，整体延迟较大

可选的转发方案

splice：
- 基于管道pipe机制，将一个socket的接收通过管道重定向到另一个socket的发送，避免用户层的数据拷贝
- 不适用一对多的转发

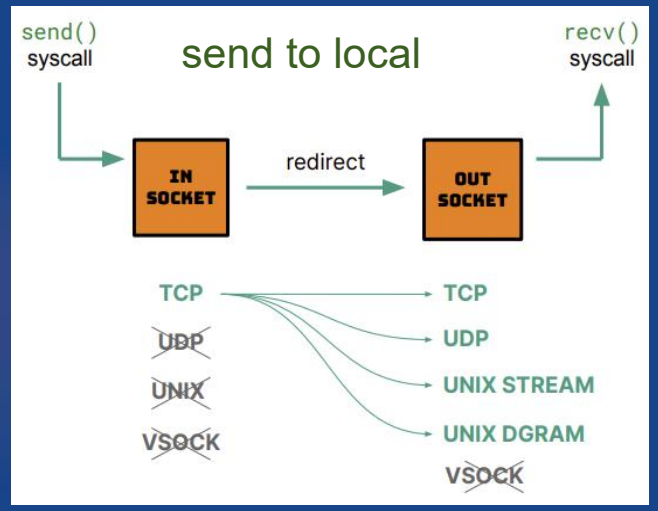
DPDK：
- 旁路内核协议栈，用户态程序直接从网卡收发数据
- 整个转发系统需要重构，改造代价较高

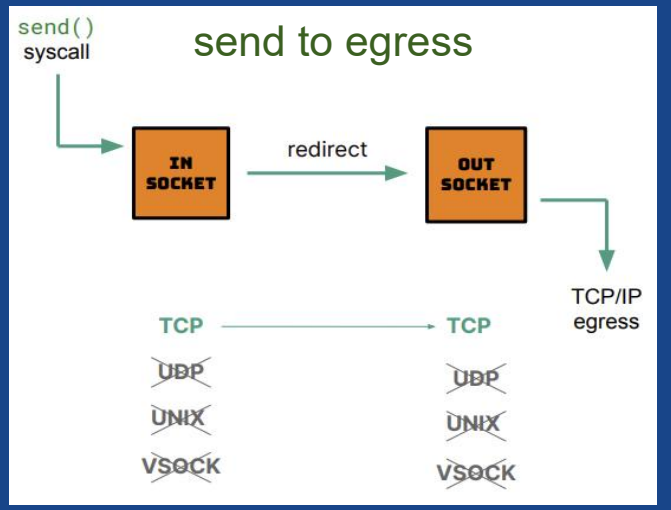
XDP：
- 在网卡驱动层直接进行数据包的转发，旁路内核协议栈
- 无法实现协议层的转换

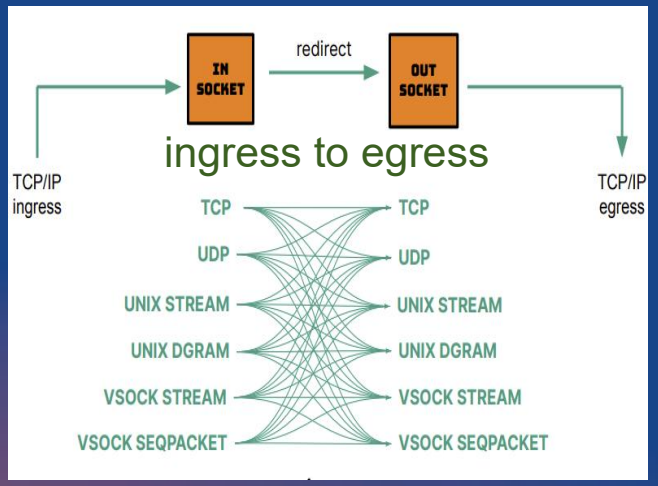
ebpf:
- 基于sockmap实现数据流重定向，旁路用户进程的转发处理
- 目前只支持一对一转发

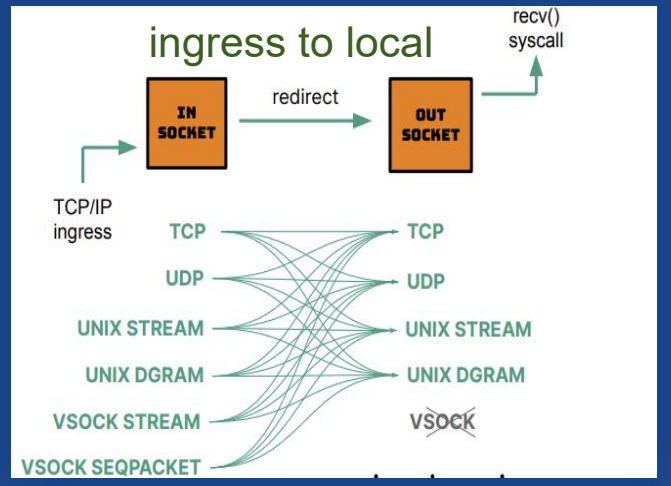
sockmap概述

sockmap自从linux 4.14版本引入后，支持的类型和特性不断拓展和延伸，其具备在内核socket层直接实现数据流的重定向，优化数据链路的流向，广泛应用于云原生、kubernetes以及服务网格等场景，加速网络传输。



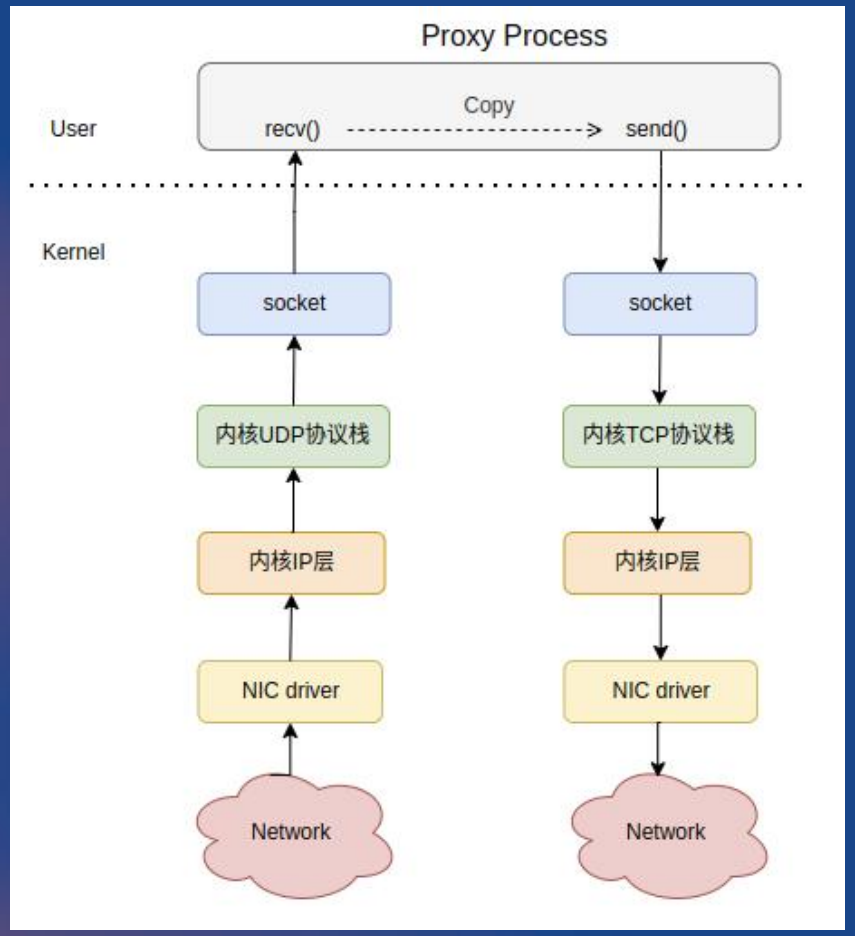




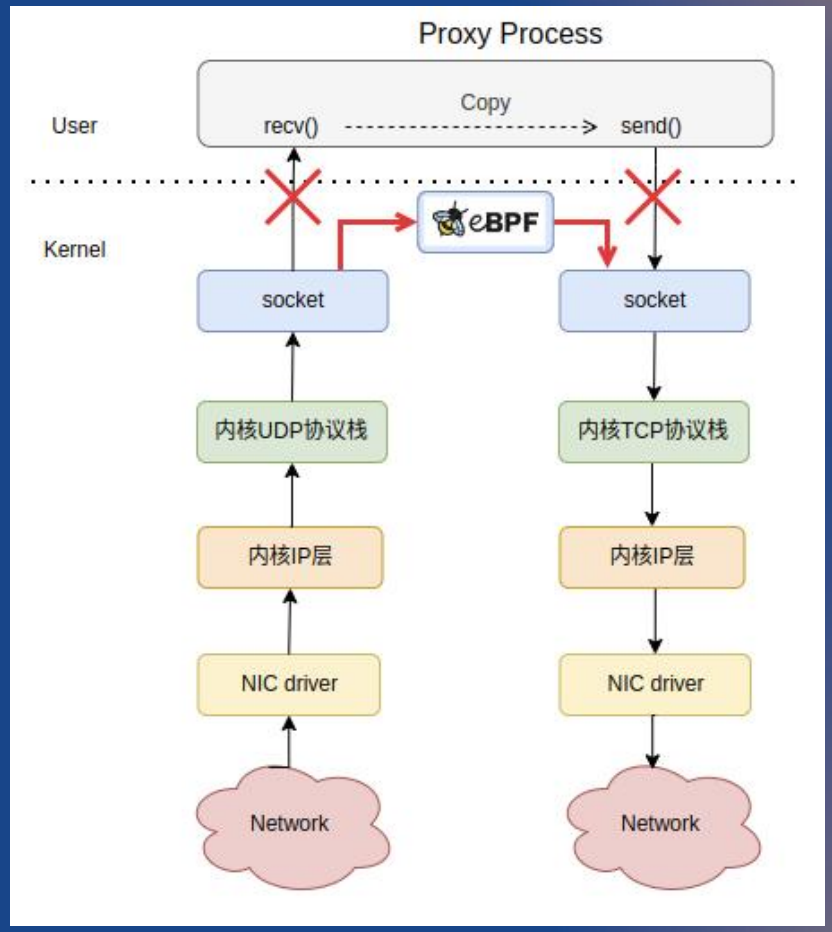


来源：https://www.usenix.org/system/files/srecon23emea-slides_sitnicki.pdf (Jakub Sitnicki, Cloudflare Inc.)

proxy和sockmap转发对比

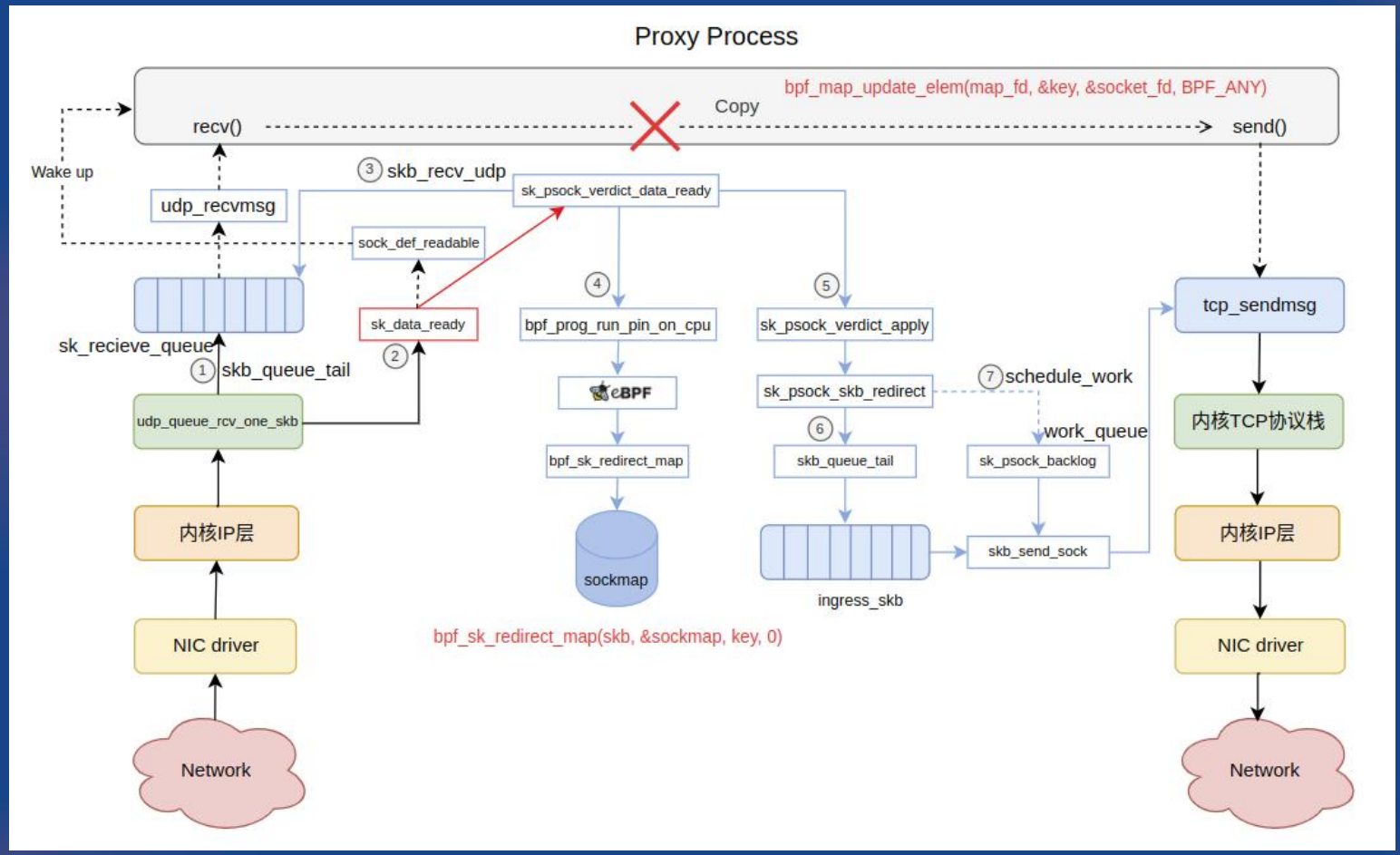


数据在内核态和用户态之间多次拷贝
频繁的系统调用，上下文切换
需要唤醒用户进程收包



sockmap直接在内核socket层实现转发
数据只在内核层拷贝，无需用户进程参与

sockmap转发路径分析



proxy： 需要唤醒用户进程去收包，数据包从内核态拷贝到用户态，再从用户态拷贝到内核态

sockmap：通过ebpf程序实现数据流的重定向（redirect），将数据包的转发流程offload到内核处理
只支持一对一转发，用户进程无法感知转发统计信息

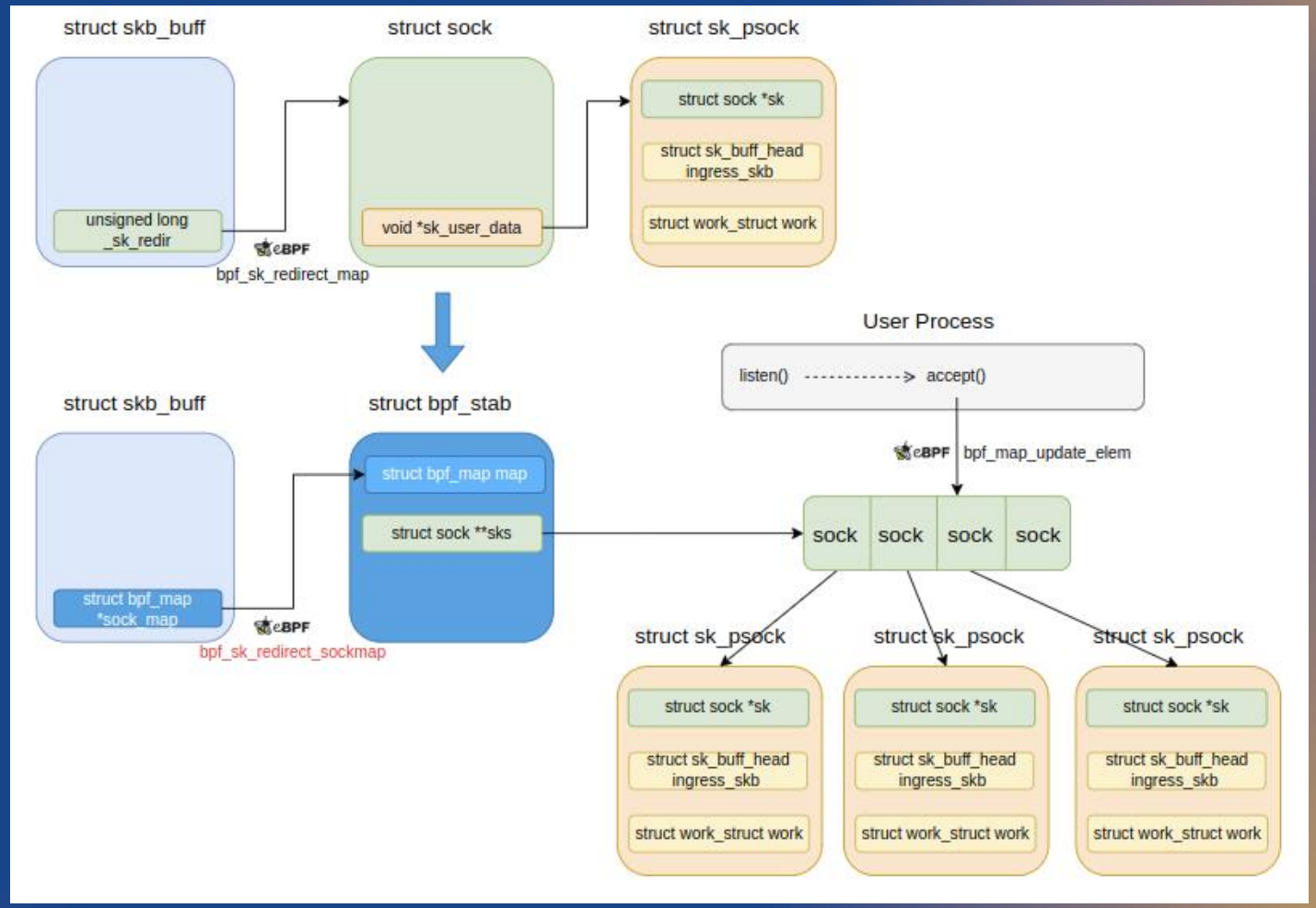
sockmap代理转发方案改进

方案基本思路：

- skb_buff结构体新增成员struct bpf_map *sock_map，记录要转发的列表
- 新增bpf helper bpf_sk_redirect_sockmap，支持一对多的转发
- 用户空间通过netlink机制订阅转发统计及状态信息

方案具体实施步骤：

- 用户进程监听客户端连接，调用ebpf接口将已建立的连接加入到sockmap转发列表
- 在内核收包流程中执行ebpf程序，调用bpf_sk_redirect_sockmap，给skb->sock_map赋值
- 在sk_psock_verdict_apply中遍历sockmap中加入的所有sock，并将数据包skb_clone后发送到对应的sock
- 如果有连接主动断开，内核通过netlink机制通知到用户进程，用户进程将对应连接从sockmap列表中删除，动态管理转发列表



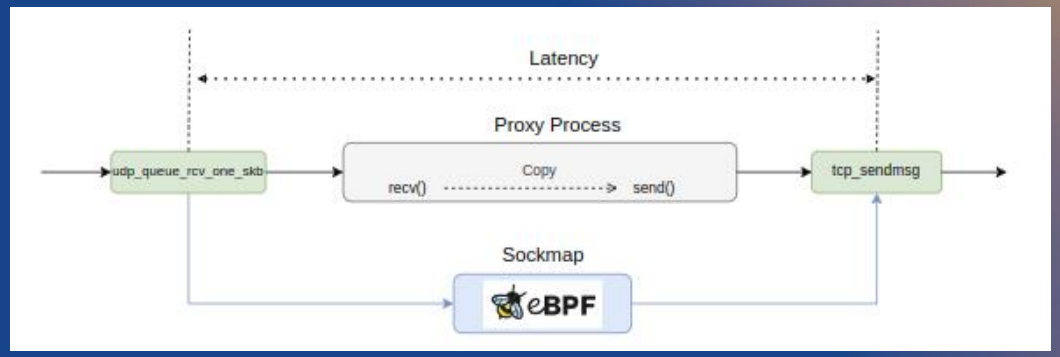
改进方案实践

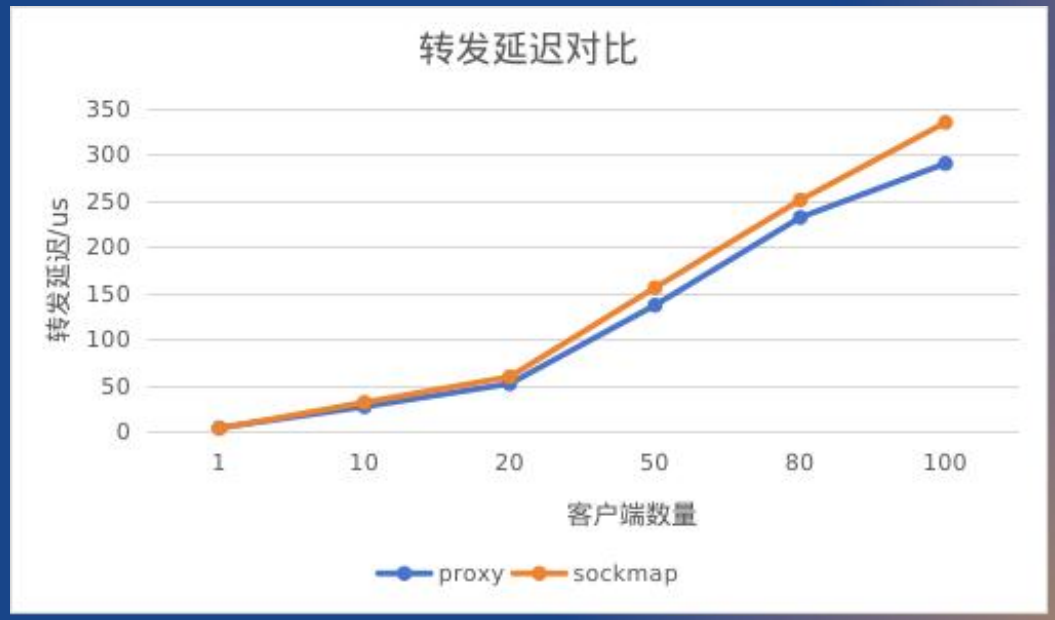
转发处理延迟：
- 计算从udp接收数据包插入缓冲区队列，到转发给tcp连接这部分的延迟

实测结果：
- sockmap方案并没有减小延迟，反而会稍微增大转发延迟

原因分析：
- 在客户端连接较多的情况下，sockmap遍历所有的sock并执行schedule_work也需要消耗一定时间
- schedule_work默认会选择当前CPU的工作队列，实际需要等软中断退出后才能执行这个工作队列，导致累积延迟
- work_queue的机制本身也会有一定的延迟

优化措施：
- schedule_work改为queue_work_on，并指定到不同CPU上，调度多个工作队列并发执行





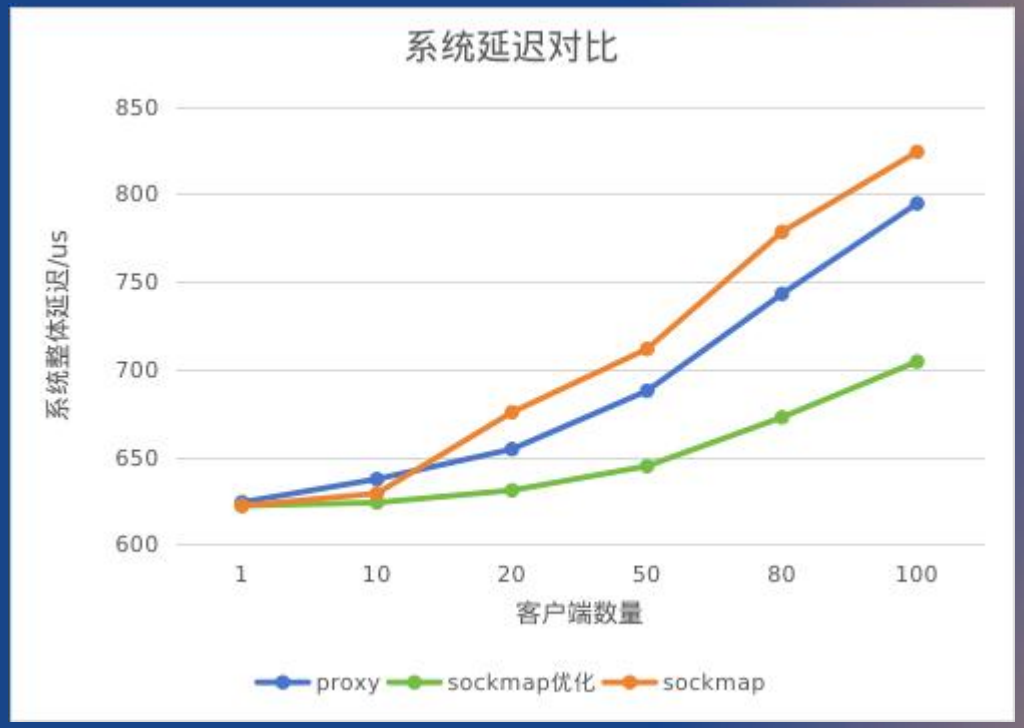
优化效果

系统整体延迟：

- 在数据源发送数据时打上时间戳，客户端接收到数据时计算与发送时间戳的差值
- 对所有客户端计算得到的延迟取平均

实测优化效果：

- 采用多个工作队列并发执行带来的优化效果明显，**系统整体平均延迟下降10%左右**



方案总结与后续展望

sockmap代理转发方案实践总结：

- 旁路用户进程的数据转发处理，在内核socket层实现一对多的转发，减少系统调用和数据的来回拷贝
- 用户进程通过netlink接口查询转发统计及状态信息，动态管理转发列表
- sockmap单工作队列机制并不能降低延迟，采用多工作队列并发才有一定优化效果

后续展望：

- 针对一对多的转发场景，进一步优化sockmap的转发执行路径
- 继续完善方案，增加容错和异常处理

THANKS

<luyun@kylinos.cn>