第三届 eBPF开发者大会

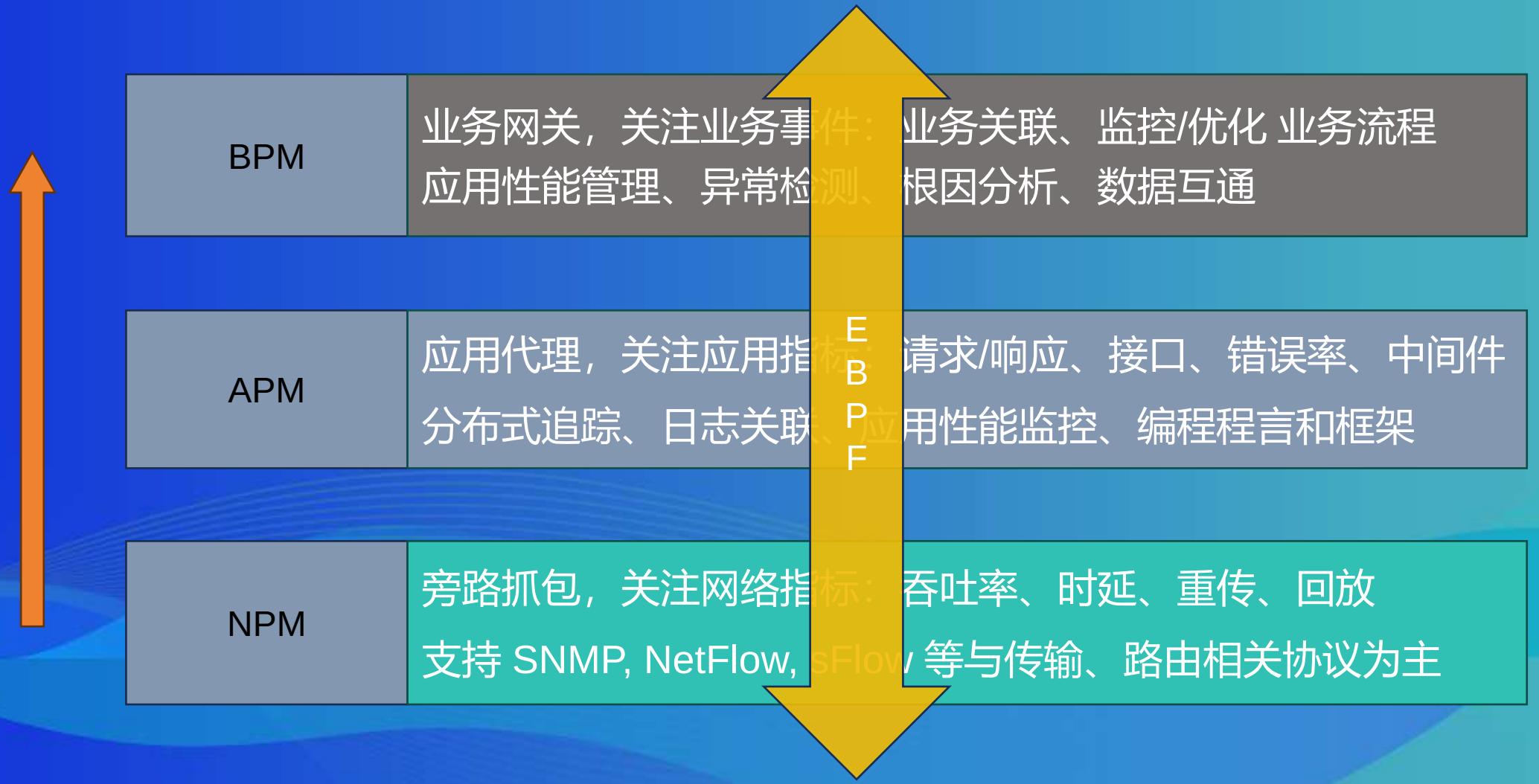
www.ebpftravel.com

eBPF 在故障定位的应用

匠心

杭州乘云数字有限公司

中国·西安


BPM
业务网关，关注业务事件：业务关联、监控/优化 业务流程
应用性能管理、异常检测、根因分析、数据互通
APM
应用代理，关注应用指标：请求/响应、接口、错误率、中间件
分布式追踪、日志关联、应用性能监控、编程程言和框架
NPM
旁路抓包，关注网络指标：吞吐率、时延、重传、回放
支持 SNMP, NetFlow, sFlow 等与传输、路由相关协议为主
EBPF

eBPF + APM 为我们带来的核心能力

- 更精准的数据采集，脱离“全流量”的苦海
- 更丰富的指标维度，如pid, offset
- 更精准的指标计算，如rtt
- 更多的指标类型，如 tcp queue

eBPF + APM 为我们带来的核心能力

- 将“微服务”的排障粒度从接口延伸到“访问资源”。
- 更便捷的应用/业务洞察能力
- 事后分析 -> 事前预警 / 事中告警

故障案例 1

- Nagle + 延迟确认

No.	Time	Source	SrcPort	Destination	DstPort	Protocol	Length	Info
1	0.000000	223.104.213.23	15833	172.17.0.9	8990	TCP	78	15833 → 8990 [SYN] Seq=0 Win=65535 Len=0 MSS=1400 WS=64 TSval=304380088 TSecr=0 SACK
2	0.000033	172.17.0.9	8990	223.104.213.23	15833	TCP	74	8990 → 15833 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=1591016
3	0.029843	223.104.213.23	15833	172.17.0.9	8990	TCP	66	15833 → 8990 [ACK] Seq=1 Ack=1 Win=131840 Len=0 TSval=304380125 TSecr=159101656
4	0.009957	223.104.213.23	15833	172.17.0.9	8990	HTTP	426	GET / HTTP/1.1
5	0.000025	172.17.0.9	8990	223.104.213.23	15833	TCP	66	8990 → 15833 [ACK] Seq=1 Ack=361 Win=30080 Len=0 TSval=159101696 TSecr=304380125
6	0.141651	172.17.0.9	8990	223.104.213.23	15833	HTTP/JSON	309	HTTP/1.1 200 OK , JSON (application/json)
7	0.038302	223.104.213.23	15833	172.17.0.9	8990	TCP	66	15833 → 8990 [ACK] Seq=361 Ack=244 Win=131584 Len=0 TSval=304380295 TSecr=159101838
8	3.080154	223.104.213.23	15833	172.17.0.9	8990	HTTP	426	GET / HTTP/1.1
9	0.000026	172.17.0.9	8990	223.104.213.23	15833	TCP	66	8990 → 15833 [ACK] Seq=244 Ack=721 Win=31104 Len=0 TSval=159104956 TSecr=304383319
10	0.001140	172.17.0.9	8990	223.104.213.23	15833	HTTP/JSON	309	HTTP/1.1 200 OK , JSON (application/json)
11	0.038643	223.104.213.23	15833	172.17.0.9	8990	TCP	66	15833 → 8990 [ACK] Seq=721 Ack=487 Win=131328 Len=0 TSval=304383373 TSecr=159104957
12	1.760097	223.104.213.23	15833	172.17.0.9	8990	HTTP	426	GET / HTTP/1.1
13	0.000980	172.17.0.9	8990	223.104.213.23	15833	HTTP/JSON	309	HTTP/1.1 200 OK , JSON (application/json)
14	0.038936	223.104.213.23	15833	172.17.0.9	8990	TCP	66	15833 → 8990 [ACK] Seq=1081 Ack=730 Win=131072 Len=0 TSval=304385145 TSecr=159106757
15	1.690055	223.104.213.23	15833	172.17.0.9	8990	HTTP	426	GET / HTTP/1.1
16	0.000932	172.17.0.9	8990	223.104.213.23	15833	HTTP/JSON	309	HTTP/1.1 200 OK , JSON (application/json)
17	0.039044	223.104.213.23	15833	172.17.0.9	8990	TCP	66	15833 → 8990 [ACK] Seq=1441 Ack=973 Win=130880 Len=0 TSval=304386863 TSecr=159108487

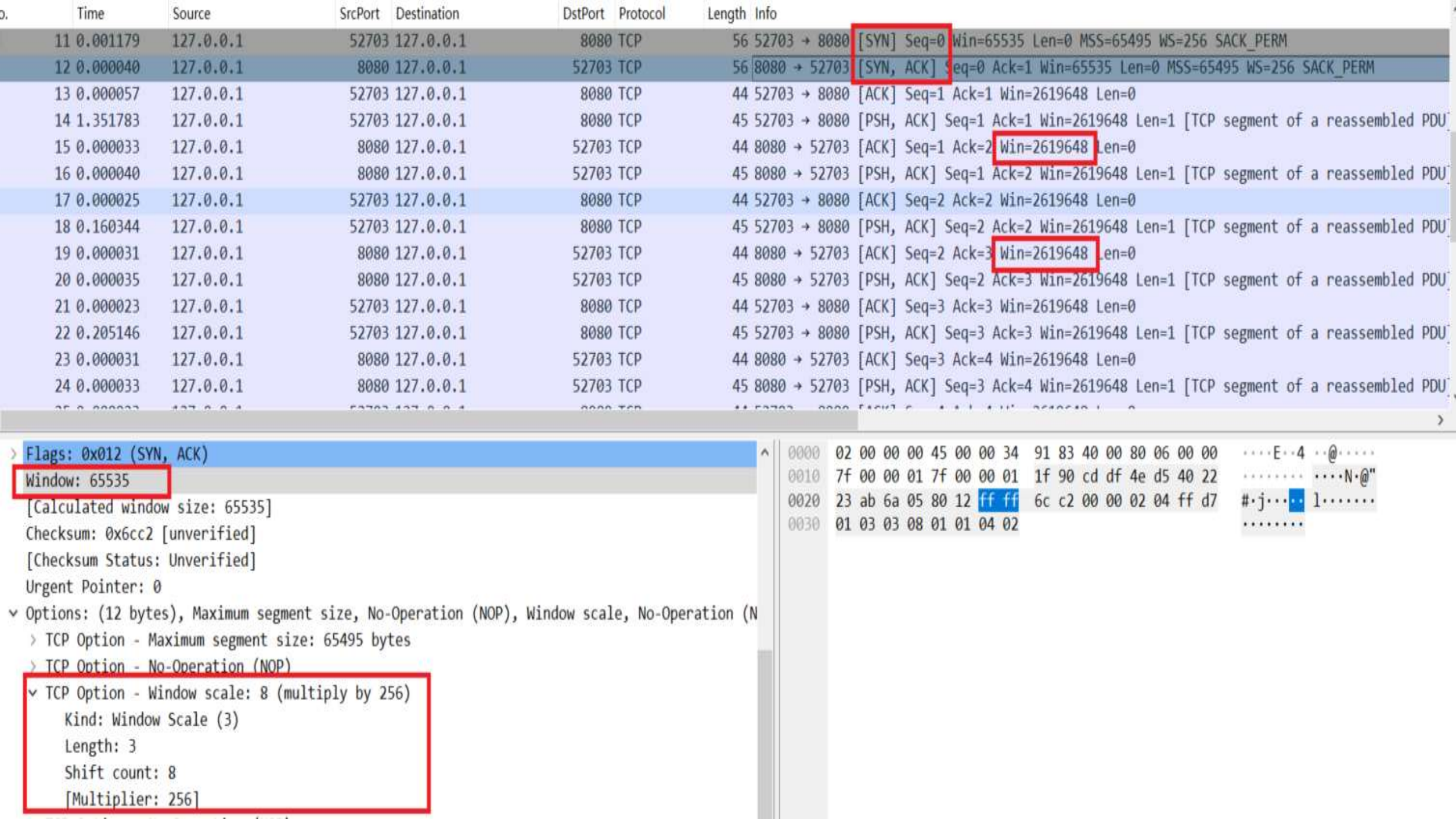
故障案例 2

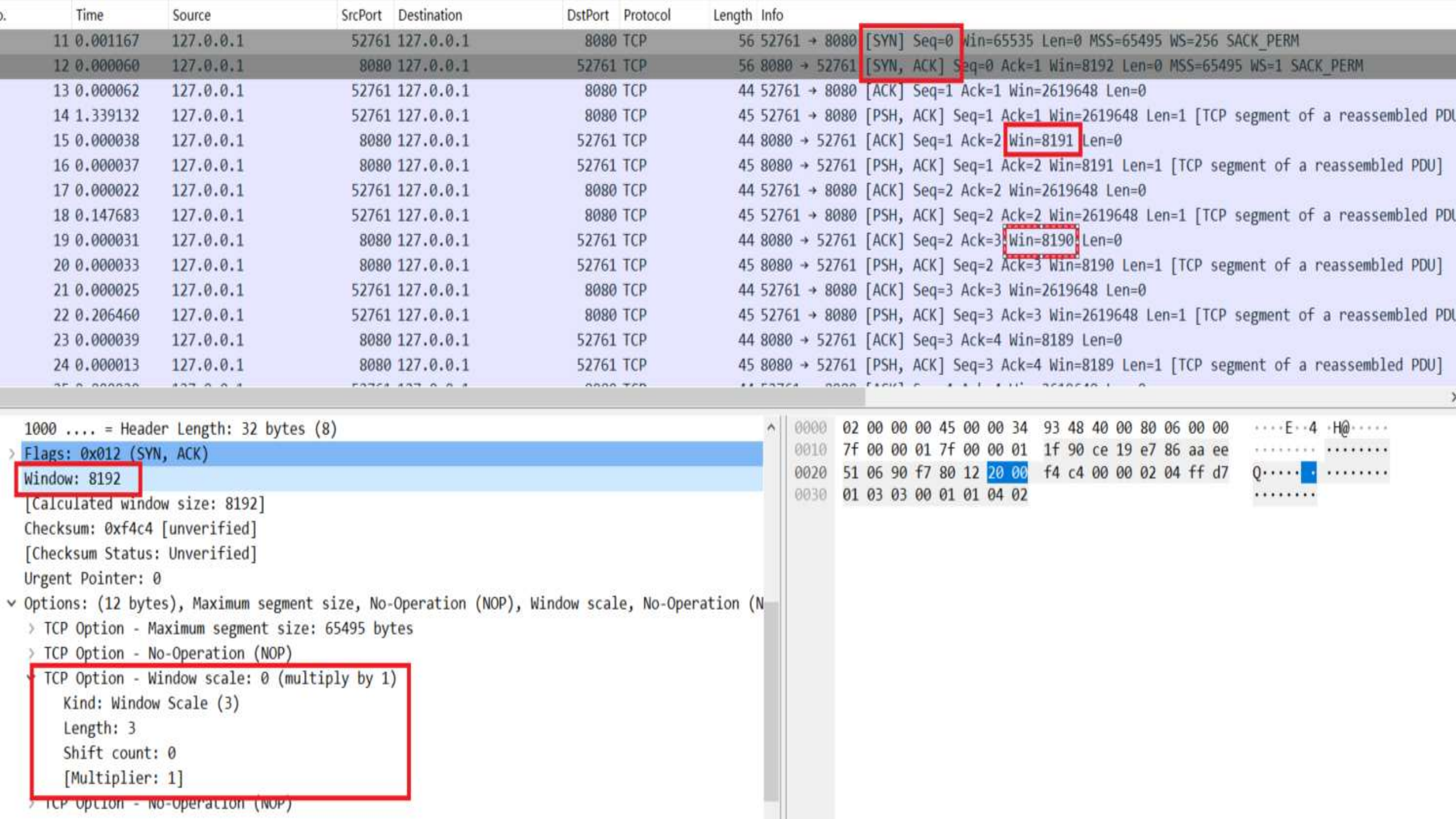
- 错误的 socket 选项设置

```
// 开始监听
listen(server_socket, 3);
printf("Server listening...\n");

// 接受传入连接
client_len = sizeof(struct sockaddr_in);
client_socket = accept(server_socket, (struct sockaddr*)&client, &client_len);
if (client_socket == INVALID_SOCKET) {
    printf("Accept failed. Error Code: %d\n", WSAGetLastError());
    closesocket(server_socket);
    WSACleanup();
    return 1;
}

// 设置接收缓冲区大小为 8K
int opt_val = RECV_BUFFER_SIZE;
if (setsockopt(client_socket, SOL_SOCKET, SO_RCVBUF, (char*)&opt_val, sizeof(opt_val)) == SOCKET_ERROR) {
    printf("setsockopt(SO_RCVBUF) failed. Error Code: %d\n", WSAGetLastError());
    closesocket(server_socket);
    WSACleanup();
    return 1;
}
```


No. Time Source SrcPort Destination DstPort Protocol Length Info
11 0.001179 127.0.0.1 52703 127.0.0.1 8080 TCP 56 52703 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
12 0.000040 127.0.0.1 8080 127.0.0.1 52703 TCP 56 8080 → 52703 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
13 0.000057 127.0.0.1 52703 127.0.0.1 8080 TCP 44 52703 → 8080 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
14 1.351783 127.0.0.1 52703 127.0.0.1 8080 TCP 45 52703 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=1 [TCP segment of a reassembled PDU
15 0.000033 127.0.0.1 8080 127.0.0.1 52703 TCP 44 8080 → 52703 [ACK] Seq=1 Ack=2 Win=2619648 Len=0
16 0.000040 127.0.0.1 8080 127.0.0.1 52703 TCP 45 8080 → 52703 [PSH, ACK] Seq=1 Ack=2 Win=2619648 Len=1 [TCP segment of a reassembled PDU
17 0.000025 127.0.0.1 52703 127.0.0.1 8080 TCP 44 52703 → 8080 [ACK] Seq=2 Ack=2 Win=2619648 Len=0
18 0.160344 127.0.0.1 52703 127.0.0.1 8080 TCP 45 52703 → 8080 [PSH, ACK] Seq=2 Ack=2 Win=2619648 Len=1 [TCP segment of a reassembled PDU
19 0.000031 127.0.0.1 8080 127.0.0.1 52703 TCP 44 8080 → 52703 [ACK] Seq=2 Ack=3 Win=2619648 Len=0
20 0.000035 127.0.0.1 8080 127.0.0.1 52703 TCP 45 8080 → 52703 [PSH, ACK] Seq=2 Ack=3 Win=2619648 Len=1 [TCP segment of a reassembled PDU
21 0.000023 127.0.0.1 52703 127.0.0.1 8080 TCP 44 52703 → 8080 [ACK] Seq=3 Ack=3 Win=2619648 Len=0
22 0.205146 127.0.0.1 52703 127.0.0.1 8080 TCP 45 52703 → 8080 [PSH, ACK] Seq=3 Ack=3 Win=2619648 Len=1 [TCP segment of a reassembled PDU
23 0.000031 127.0.0.1 8080 127.0.0.1 52703 TCP 44 8080 → 52703 [ACK] Seq=3 Ack=4 Win=2619648 Len=0
24 0.000033 127.0.0.1 8080 127.0.0.1 52703 TCP 45 8080 → 52703 [PSH, ACK] Seq=3 Ack=4 Win=2619648 Len=1 [TCP segment of a reassembled PDU
Flags: 0x012 (SYN, ACK)
Window: 65535
[Calculated window size: 65535]
Checksum: 0x6cc2 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (N
TCP Option - Maximum segment size: 65495 bytes
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 8 (multiply by 256)
Kind: Window Scale (3)
Length: 3
Shift count: 8
[Multiplier: 256]
0000 02 00 00 00 45 00 00 34 91 83 40 00 80 06 00 00 ····E··4 ··@·····
0010 7f 00 00 01 7f 00 00 01 1f 90 cd df 4e d5 40 22 ···· ···· ····N·@"
0020 23 ab 6a 05 80 12 ff ff 6c c2 00 00 02 04 ff d7 #·j····· l·····
0030 01 03 03 08 01 01 04 02 ····· ····


No. Time Source SrcPort Destination DstPort Protocol Length Info
11 0.001167 127.0.0.1 52761 127.0.0.1 8080 TCP 56 52761 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
12 0.000060 127.0.0.1 8080 127.0.0.1 52761 TCP 56 8080 → 52761 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=65495 WS=1 SACK_PERM
13 0.000062 127.0.0.1 52761 127.0.0.1 8080 TCP 44 52761 → 8080 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
14 1.339132 127.0.0.1 52761 127.0.0.1 8080 TCP 45 52761 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=1 [TCP segment of a reassembled PDU
15 0.000038 127.0.0.1 8080 127.0.0.1 52761 TCP 44 8080 → 52761 [ACK] Seq=1 Ack=2 Win=8191 Len=0
16 0.000037 127.0.0.1 8080 127.0.0.1 52761 TCP 45 8080 → 52761 [PSH, ACK] Seq=1 Ack=2 Win=8191 Len=1 [TCP segment of a reassembled PDU]
17 0.000022 127.0.0.1 52761 127.0.0.1 8080 TCP 44 52761 → 8080 [ACK] Seq=2 Ack=2 Win=2619648 Len=0
18 0.147683 127.0.0.1 52761 127.0.0.1 8080 TCP 45 52761 → 8080 [PSH, ACK] Seq=2 Ack=2 Win=2619648 Len=1 [TCP segment of a reassembled PDU
19 0.000031 127.0.0.1 8080 127.0.0.1 52761 TCP 44 8080 → 52761 [ACK] Seq=2 Ack=3 Win=8190 Len=0
20 0.000033 127.0.0.1 8080 127.0.0.1 52761 TCP 45 8080 → 52761 [PSH, ACK] Seq=2 Ack=3 Win=8190 Len=1 [TCP segment of a reassembled PDU]
21 0.000025 127.0.0.1 52761 127.0.0.1 8080 TCP 44 52761 → 8080 [ACK] Seq=3 Ack=3 Win=2619648 Len=0
22 0.206460 127.0.0.1 52761 127.0.0.1 8080 TCP 45 52761 → 8080 [PSH, ACK] Seq=3 Ack=3 Win=2619648 Len=1 [TCP segment of a reassembled PDU
23 0.000039 127.0.0.1 8080 127.0.0.1 52761 TCP 44 8080 → 52761 [ACK] Seq=3 Ack=4 Win=8189 Len=0
24 0.000013 127.0.0.1 8080 127.0.0.1 52761 TCP 45 8080 → 52761 [PSH, ACK] Seq=3 Ack=4 Win=8189 Len=1 [TCP segment of a reassembled PDU]
1000 .... = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
Window: 8192
[Calculated window size: 8192]
Checksum: 0xf4c4 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (N
TCP Option - Maximum segment size: 65495 bytes
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 0 (multiply by 1)
Kind: Window Scale (3)
Length: 3
Shift count: 0
[Multiplier: 1]
0000 02 00 00 00 45 00 00 34 93 48 40 00 80 06 00 00 ····E··4 ·H@·····
0010 7f 00 00 01 7f 00 00 01 1f 90 ce 19 e7 86 aa ee ···· ····
0020 51 06 90 f7 80 12 20 00 f4 c4 00 00 02 04 ff d7 Q····· ····
0030 01 03 03 00 01 01 04 02 ····

BPF_CGROUP_SOCK_OPS

BPF_CGROUP_SOCK_OPS 程序附加到 cgroup 上，会在以下 socket 事件发生时被调用：

- 连接建立 (TCP)
- 连接终止
- 数据发送/接收
- 重传超时
- 拥塞控制状态变化等

kprobe/tcp_sendmsg, kretprobe/tcp_recvmsg

- 以流为单位，建立指标体系
  - 五元组
  - 进程ID, 进程名
  - 收/发包数, 收/发字节数, 重传包数 [Counter]
  - Rtt / Rtt-var [Histogram]
  - 连接状态 [Gauge]
  - RWND [Histogram]

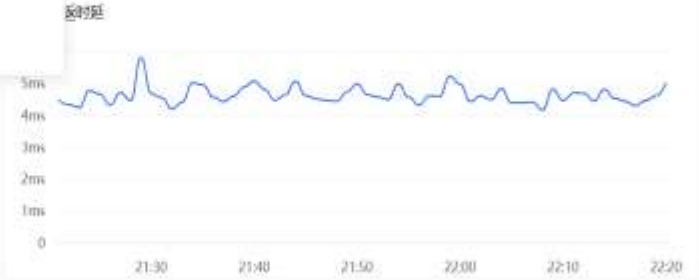
- OneAgent 辅助，尽量减少数据量

客户端视角: host
服务端视角: host

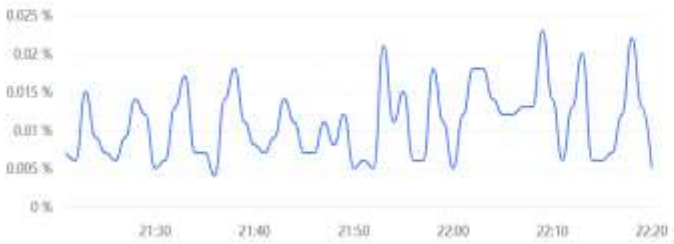
OR

host
cname
host
ip
pod_name
port
svc_name

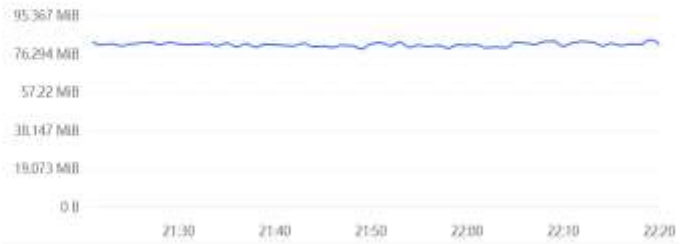
图表分析



TCP传输重传百分比



网络发送流量



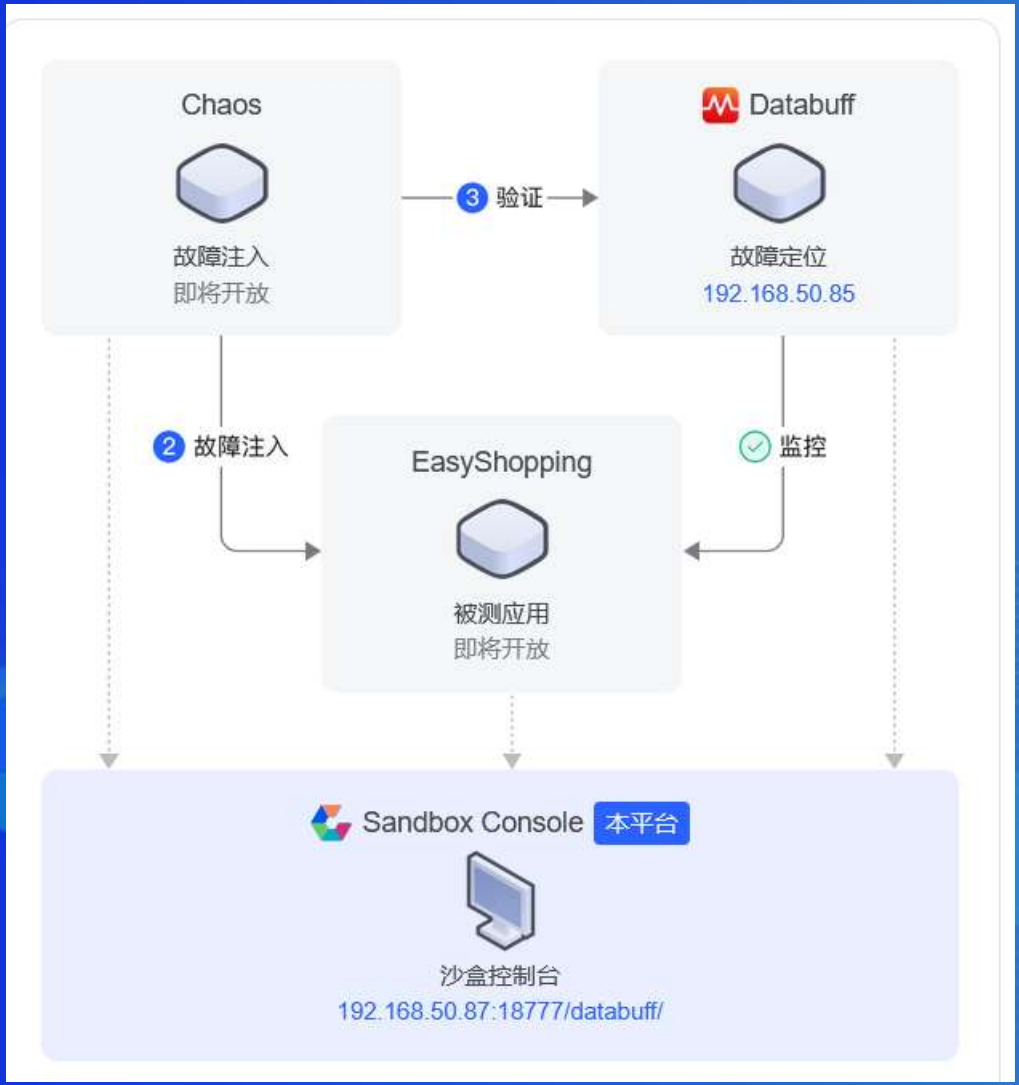
客户端	服务端	客户端→服务端 流量	服务端→客户端 流量	TCP 重传次数	延迟	RTT	抖动
k8s-b1	-	1.06 GiB / 18.37 MiB/s	723.49 MiB / 12.06 MiB/s	189 req	2.76 s	5.51 s	3.92 s
k8s-b1	host86	501.83 MiB / 8.36 MiB/s	15.2 MiB / 259.49 KiB/s	345 req	1.09 s	2.18 s	814.25 ms
k8s-b1	k8s-b1	208.93 MiB / 3.48 MiB/s	394.49 MiB / 6.57 MiB/s	87 req	3.2 s	6.4 s	6.72 s
k8s-b1	k8s-b2	4.18 MiB / 71.39 KiB/s	338.53 KiB / 5.64 KiB/s	0	2.4 s	4.8 s	8.87 s
k8s-b1	k8s-b3	877.42 KiB / 14.62 KiB/s	33.44 KiB / 570.75 B/s	0	114 ms	228 ms	22 ms
k8s-b1	k8s-b4	4.14 MiB / 70.62 KiB/s	334.73 KiB / 5.56 KiB/s	0	2.6 s	5.2 s	9.41 s
k8s-b1	k8s-b5	4.08 MiB / 69.6 KiB/s	324.45 KiB / 5.41 KiB/s	0	1.85 s	3.7 s	6.66 s
k8s-b2	-	354.35 MiB / 5.91 MiB/s	474.75 MiB / 7.91 MiB/s	77 req	2.84 s	5.68 s	6.28 s
k8s-b2	host86	162.66 MiB / 2.71 MiB/s	7.66 MiB / 130.78 KiB/s	168 req	389.5 ms	779 ms	607.5 ms
k8s-b2	k8s-b2	94.4 MiB / 1.57 MiB/s	185.41 MiB / 3.09 MiB/s	4 req	3.94 s	7.89 s	8.06 s
k8s-b3	-	1.2 GiB / 20.56 MiB/s	274.55 MiB / 4.58 MiB/s	204 req	3.68 s	7.35 s	8.35 s
k8s-b3	host86	621.2 MiB / 10.35 MiB/s	13.8 MiB / 235.58 KiB/s	447 req	242.25 ms	484.5 ms	315.5 ms
k8s-b3	k8s-b1	1.03 MiB / 17.6 KiB/s	3.02 MiB / 51.53 KiB/s	0	3.95 s	7.89 s	8.73 s
k8s-b3	k8s-b3	222.8 MiB / 3.71 MiB/s	120.74 MiB / 2.01 MiB/s	3 req	4.05 s	8.1 s	8.78 s
k8s-b4	-	155.08 MiB / 2.58 MiB/s	171.49 MiB / 2.86 MiB/s	46 req	3.34 s	6.68 s	8.06 s

port
42244
43576
42246
42242
43574
55560
56896
54232
pod_name
svc_name
cluster_id
ip_type
intra_host
cid
hostname
svc_instance
network_family
cname
svc_id
protocols
ip
direction
deployment

实时调整参数，优化传输性能

```c
SEC("sockops")
int bpf_tcp_tuning(struct bpf_sock_ops *skops)
{
    switch (skops->op) {
    case BPF_SOCK_OPS_TCP_CONNECT_CB:
        // 设置初始拥塞窗口
        bpf_setsockopt(skops, SOL_TCP, TCP_BPF_IW, &initial_window, sizeof(initial_window));
        break;
    case BPF_SOCK_OPS_RTT_CB:
        // 根据 RTT 调整参数
        if (skops->srtt_us > 200000) { // RTT > 200ms
            int mss = 536; // 减小 MSS
            bpf_setsockopt(skops, SOL_TCP, TCP_BPF_MSS, &mss, sizeof(mss));
        }
        break;
    }
    return 0;
}
```

故障演练与定位沙盒



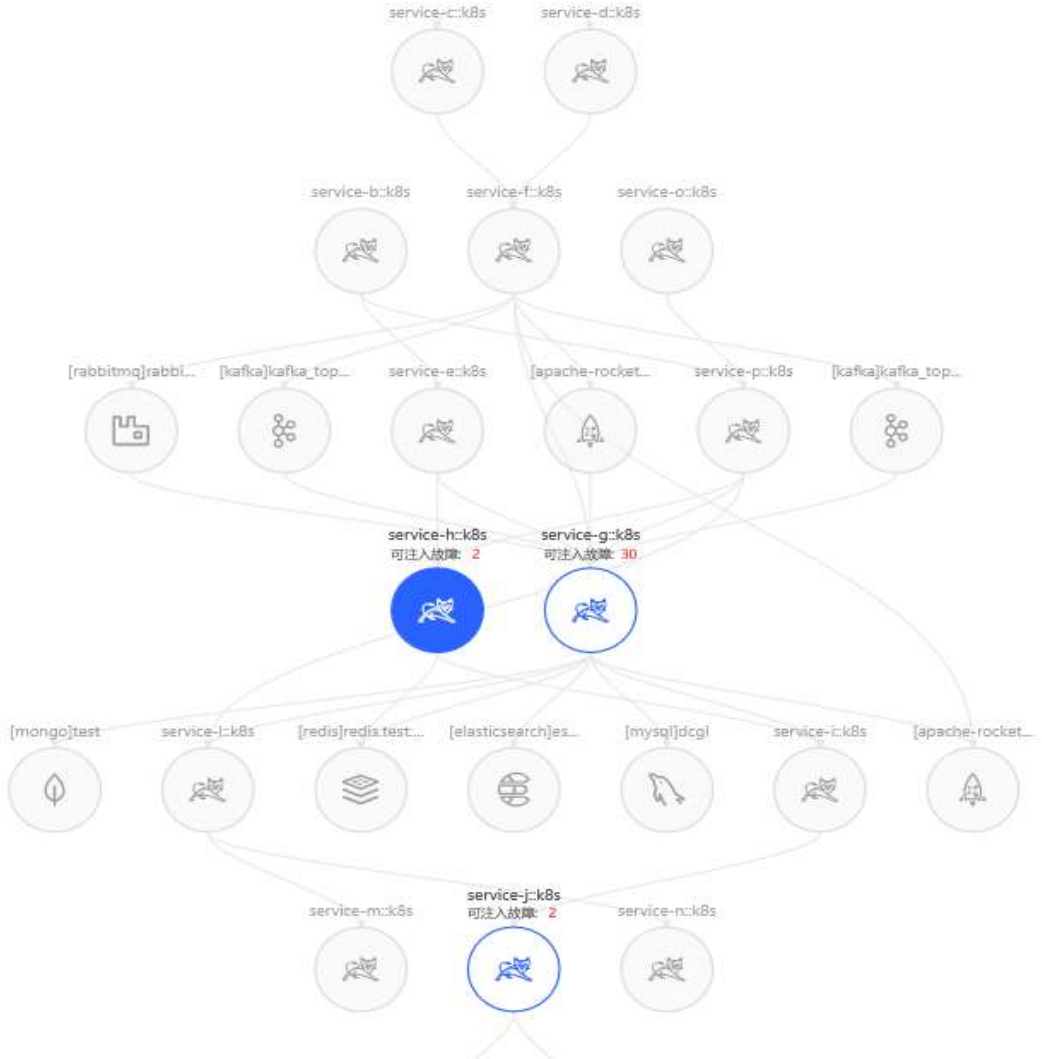
- EasyShopping：被测应用，一个包含17个服务、覆盖多种常见访问协议（Http、Dubbo、MQ、Mysql、Redis、ES、Mongo等）的复杂微服务系统。
- Chaos：故障注入平台，可以对被测应用注入各种故障。
- DataBuff：故障定位平台，监测被测应用的运行情况，当发生故障时会触发告警并自动定位根因。
- SandBox Console：沙盒控制台

故障演练与定位沙盒 – 故障注入

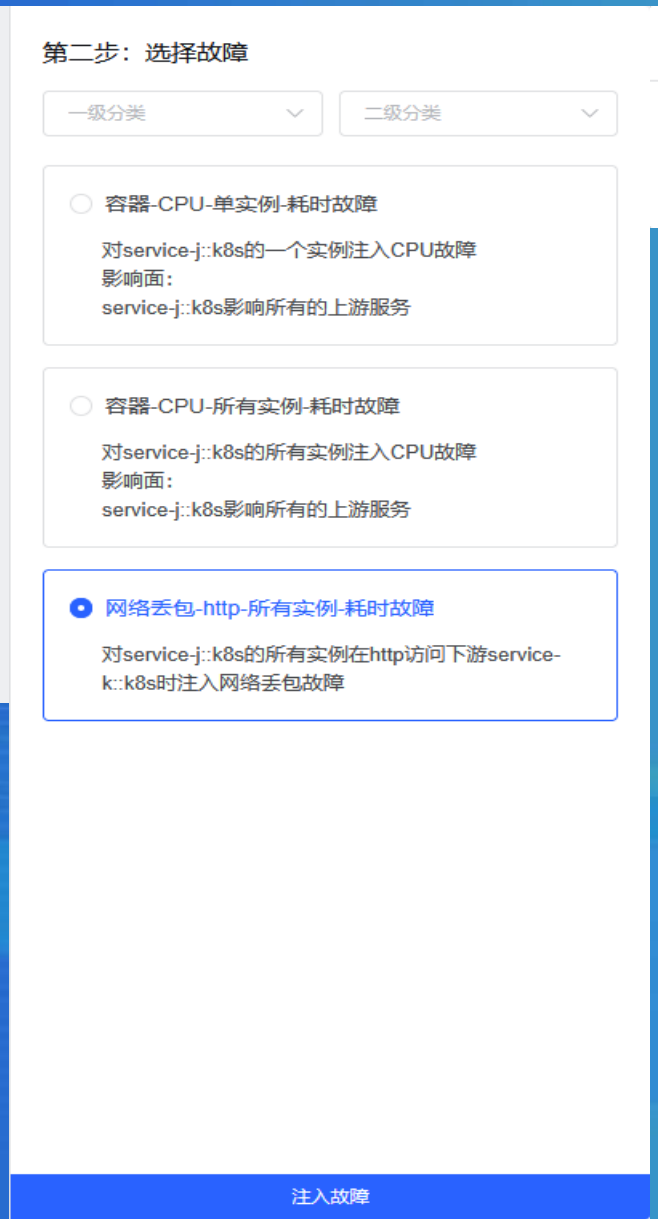
RootTalk Sandbox

- 系统介绍
- 故障注入
- 故障注入历史

第一步：选择服务节点



故障演练与定位沙盒 – 选择故障



故障演练与定位沙盒 – 查看故障

发现 2 条数据

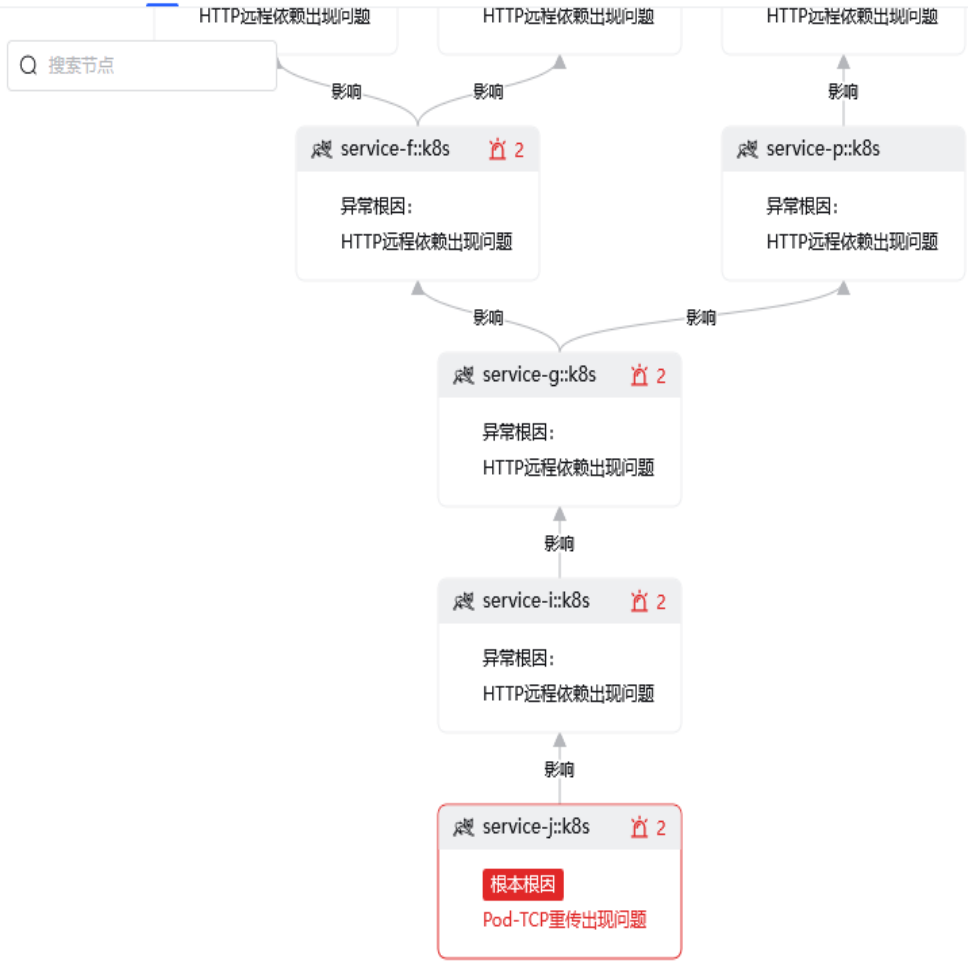
注入时间	注入状态	服务名称	故障一级分类	故障二级分类	故障名称	故障描述	注入用户	Databuff故障定位
2025-04-08 09:33:00	已完成	service-j::k8s	网络故障	丢包故障	网络丢包-http-所有实例-耗时故障	对service-j::k8s的所有实例在htt...	kuangmo	查看
2025-04-08 09:17:00	已完成	service-g::k8s	DB故障	database故障	DB-客户端-database-单实例-耗...	对service-g::k8s的单个实例的某...	luocs	查看

驾驶舱
全局拓扑
数据报表
指标体系
智能告警
告警列表
通知记录
问题列表
问题分析
应用性能
基础设施
业务观测
网络性能
访问体验
日志分析
部署配置
帮助中心
kuangmo

远程服务名【service-k::k8s】，抖动类型【抖动上升】

问题ID	P2025040800000028	影响服务	8 个				
开始时间	2025-04-08 09:33	确认时间	2025-04-08 09:35	结束时间	2025-04-08 09:35	修复时间(MTTR)	3 min	响应时间(MTTA)	3 min

告警列表 影响面分析



service-j::k8s 查看告警

异常原因：服务service-j::k8s，Pod-TCP重传出现问题

导致异常的调用：

远程服务名	抖动类型	操作
service-k::k8s	抖动上升	详情

建议优先排查上述原因以定位故障

其他原因

驾驶舱
全局拓扑
数据报表
指标体系
智能告警
应用性能
基础设施
业务观测
网络性能
网络分析
网络拓扑
DNS分析
访问体验
日志分析
部署配置
帮助中心
kuangmo

客户端视角: svc_name　　服务端视角: svc_name

客户端: laddr_svc_name:service-j::k8s

AND

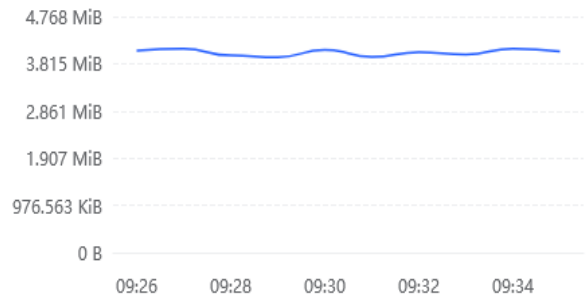
服务端: raddr_svc_name:service-k::k8s

快捷筛选

客户端 | 服务端

- port
  - 42244
  - 43576
  - 42246
  - 43578
  - 42242
  - 43574
  - 55560
  - 56896
- pod_name
- svc_name
- cluster_id
- ip_type
- intra_host
- cid
- hostname
- svc_instance
- network_family
- cname

隐藏筛选区　图表分析　设置



发现 1 条数据　设置

客户端	服务端	客户端 → 服务端 流量	服务端 → 客户端 流量	TCP 重传次数	延迟	RTT	抖动
service-j::k8s	service-k::k8s	40.4 MiB / 689.49 KiB/s	7.09 MiB / 121.06 KiB/s	1.34k req	252.83 ms	505.67 ms	158.67 ms

谢 谢！