第三届 eBPF开发者大会
www.ebpftravel.com

字节跳动基于eBPF的海量微服务高性能可观测实践

邓德杨：字节跳动系统可观测技术专家
杨腾腾：字节跳动内核网络研发工程师

中国·西安

自我介绍-邓德杨

- 字节跳动系统可观测技术专家
- eBPF可观测、混沌工程、系统诊断、性能压测等产品负责人和架构师
- 10多年工作经验，毕业后先后就职于阿里巴巴、字节跳动等
- 先后从事网络、DevOps、混沌工程、可观测及故障诊断等系统研发工作

① 背景

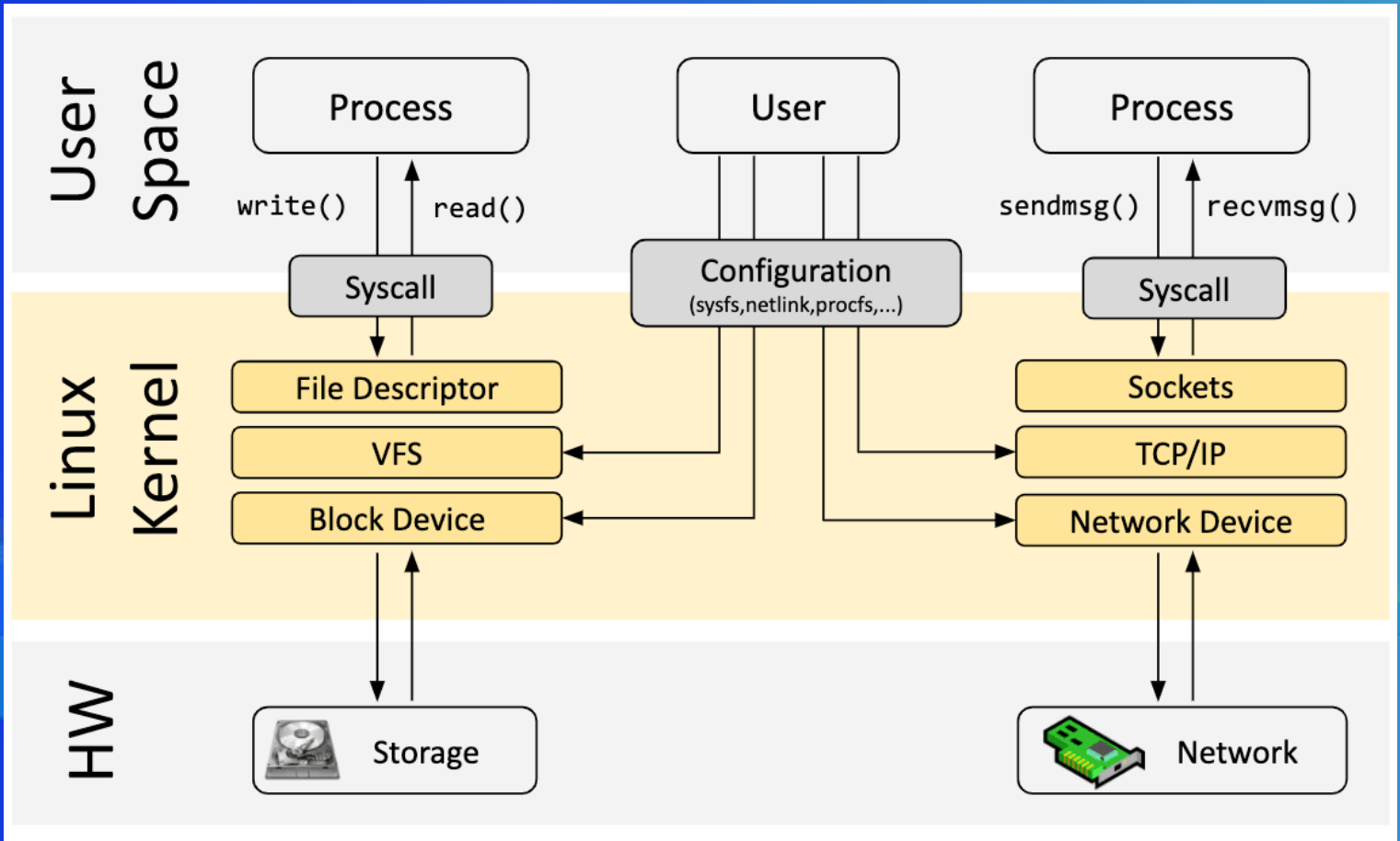
① 背景-遇到挑战

- 图中显示的是微服务调用拓扑图，每时每刻都发生海量微服务相互调用。在故障发生时，如何快速定位到故障原因，缩短MTTR(mean time to repair)，成为一个很大的挑战。

① 背景-遇到的问题

- 公司现有监控常见基于代码插桩或者SDK方式，实现数据采集
  - 接入成本高
  - 业务框架强耦合
  - 覆盖率不全
  - 链路断链

① 背景-eBPF技术



- 基于eBPF
  - 无侵入
  - 高性能
  - 可观测

① 背景-字节业务需求

- 调用链路分析->延时高/异常的服务->诊断/Profiling->Root Cause
- 机房流量治理、容灾演练
- 解决存储服务多实例间调用黑盒问题
- C++/Python等分布式服务调用链断链问题：框架推动难、性能损耗大
- 解决性能敏感的存储组件采集损耗高问题

第三届 eBPF开发者大会

www.ebpftravel.com

② 高性能可观测实践

中国·西安

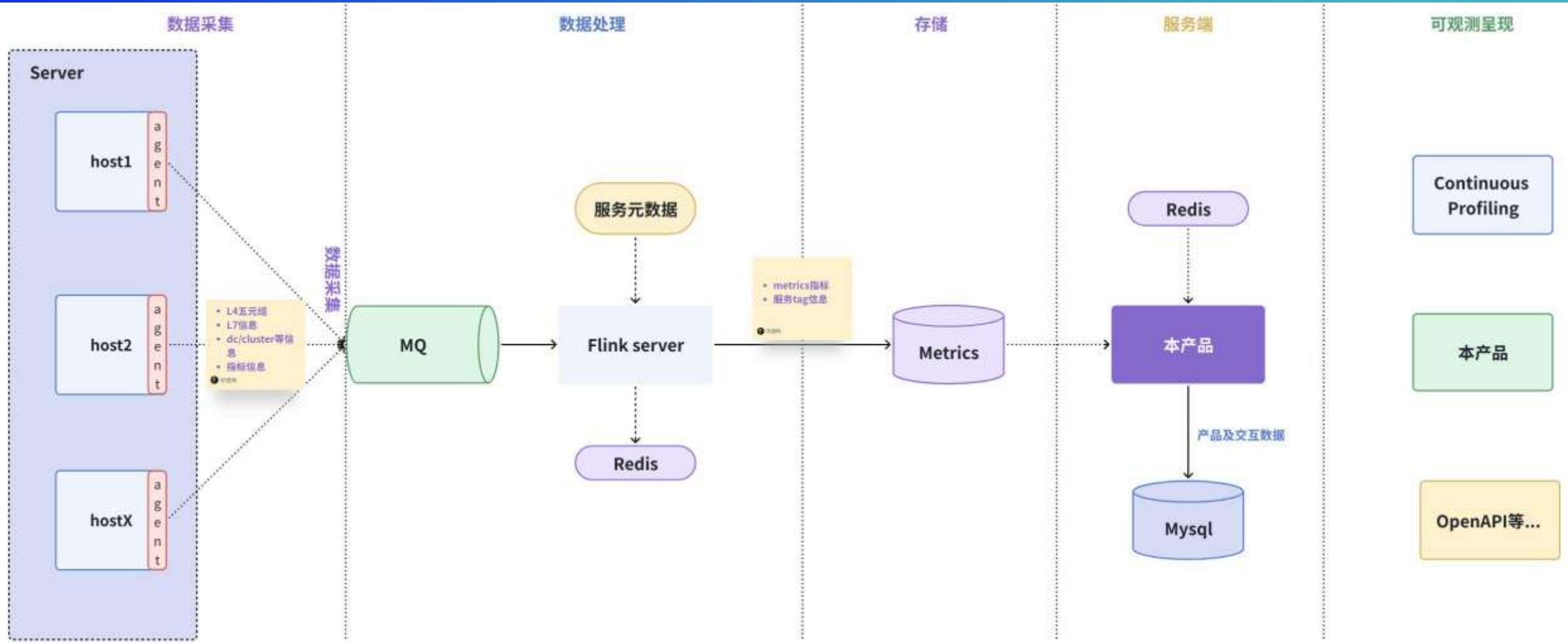
② 可观测-遇到的挑战

- 海量机器+海量微服务+海量数据
- 时间戳准确性
- 采集Pod的Id与服务映射及关联
- 性能瓶颈
- 业务多语言问题

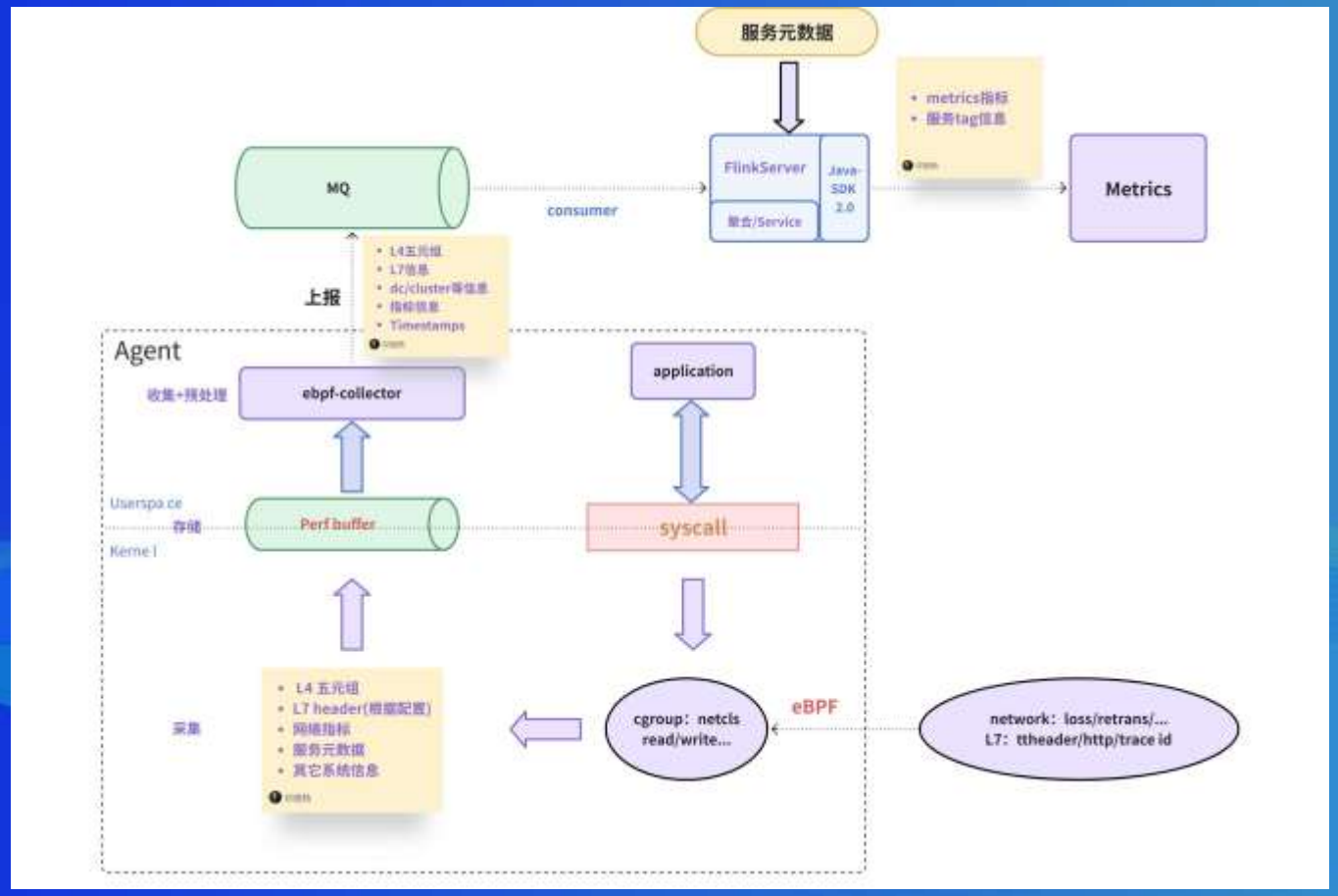
② 可观测-解决方案

- 可配置：采集+服务观测可动态配置
- 旁路解析：数据解析旁路处理，提升采集性能
- 多维度关联：服务 Id 和 Pod Id 关联
- 保留 Trace Id：使用 Trace Id+eBPF 采集，提升性能

② 可观测-系统架构


数据采集
数据处理
存储
服务端
可观测呈现
Server
host1
host2
hostX
agent
L4五元组
L7信息
dc/cluster等信息
指标信息
数据采集
MQ
服务元数据
Flink server
Redis
metrics指标
服务tag信息
Metrics
Redis
本产品
产品及交互数据
Mysql
Continuous Profiling
本产品
OpenAPI等...

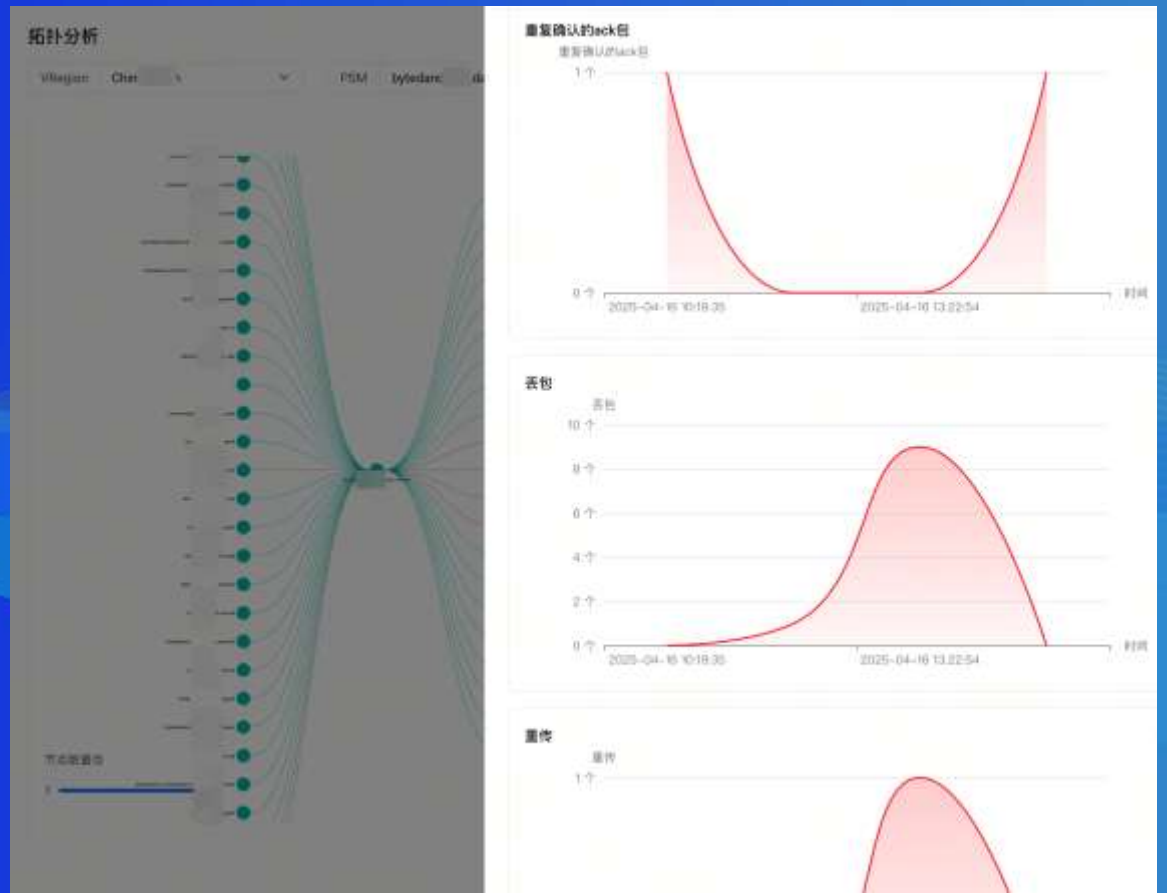
② 可观测-数据采集



- 爆炸半径控制：perfevent+agent+eBPF
- 降低采集损耗：cgroup+netcls_id
- 分布式追踪：trace_id+ttheader/http header
- 提升性能：Agent旁路梳理数据
- 元数据关联：服务id<->pod id

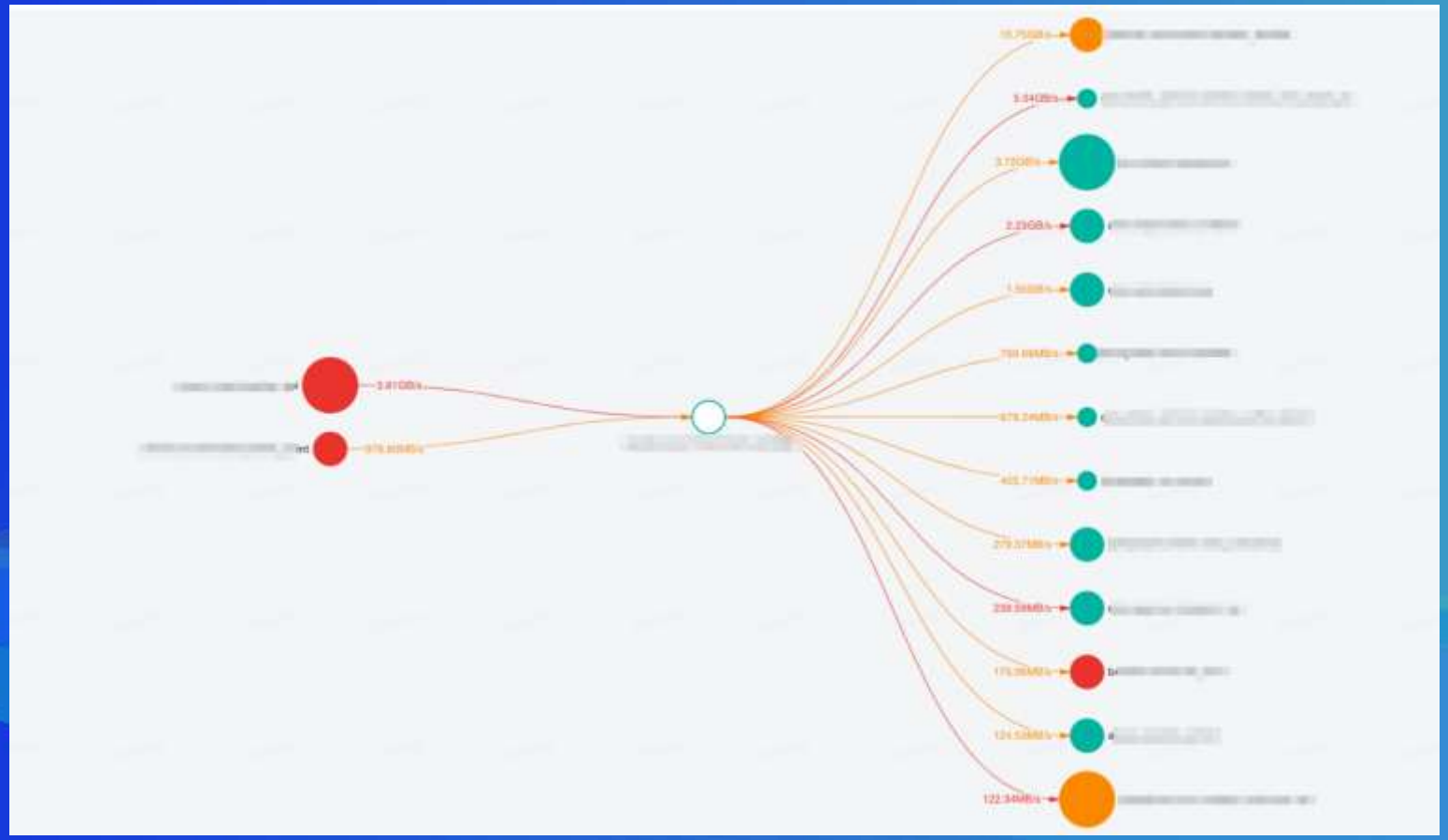
② 可观测-功能介绍

- 本产品主要基于eBPF能力，提供指标采集、网络分析、可观测、全链路分布式追踪、性能分析等能力，并打通巡检、诊断、性能分析形成全链路解决方案



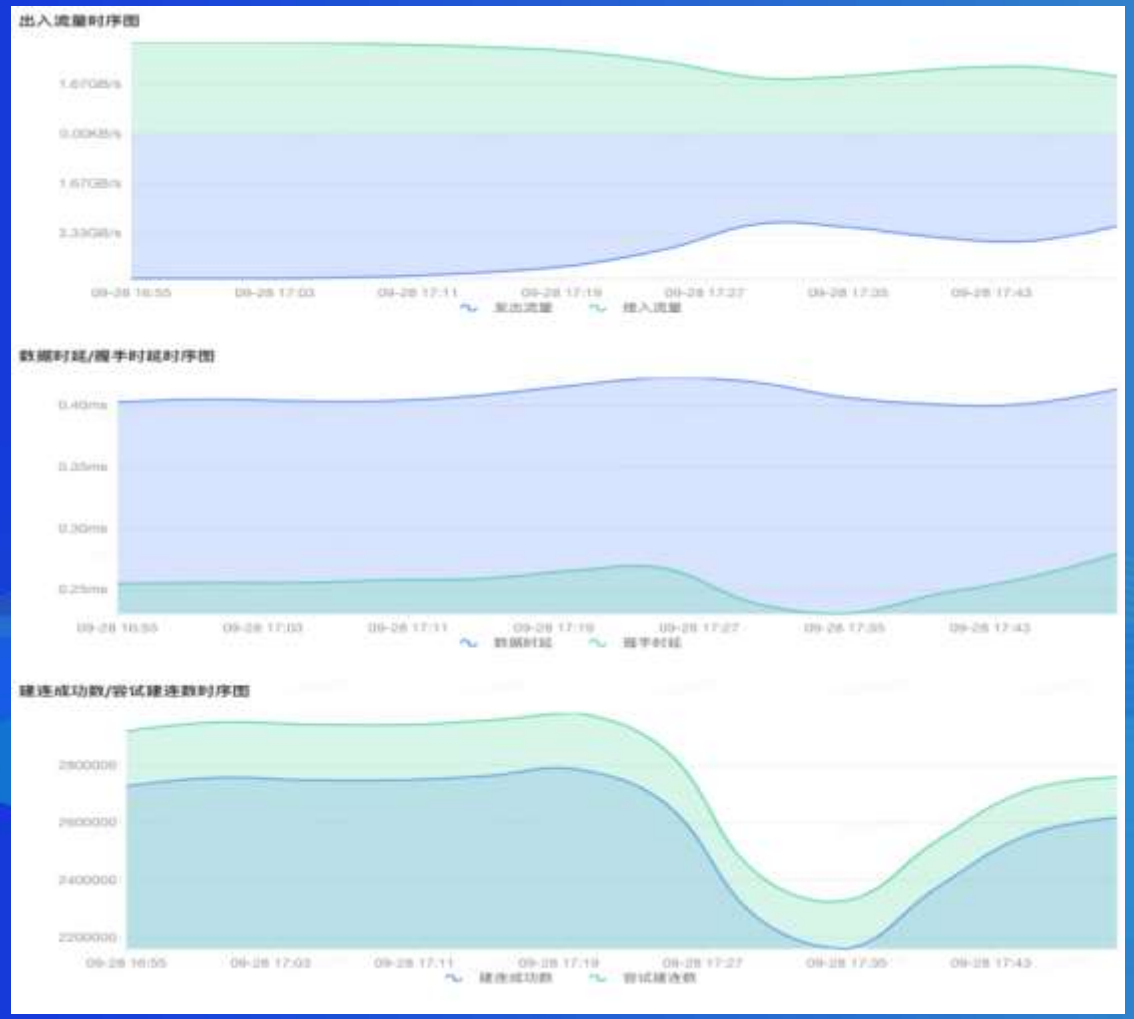
- 服务上下游异常分析
- 服务全局拓扑
- 可观测看板
- 分布式追踪拓扑
- 机房流量治理
- 可配置定制化采集

② 可观测-实践案例-上下游链路分析



- 服务上下游异常指标排查
- 链路流量观测
- 可观测看板
- 可配置定制化采集

② 可观测-实践案例-指标分析



- 出入流量时序图
- 网络丢包
- 网络重传
- 调用时延
- ……

③ 采集实现原理

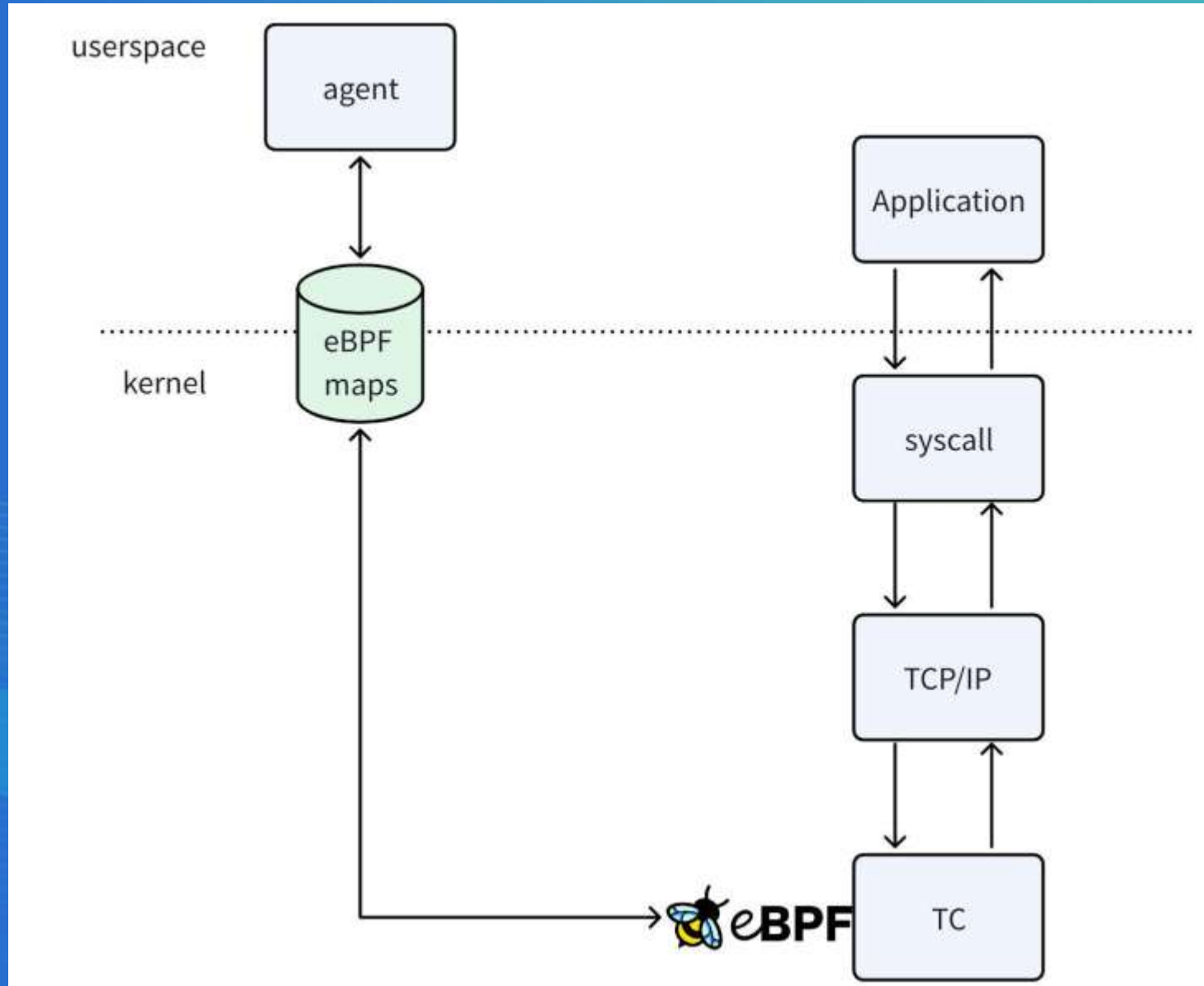
杨腾腾：字节跳动内核网络研发工程师

自我介绍-杨腾腾

- 字节跳动内核网络研发工程师
- 在内核网络领域具有丰富的经验，负责过多种网络产品的研发工作
- 深度参与eBPF在内核网络安全和网络可观测性项目

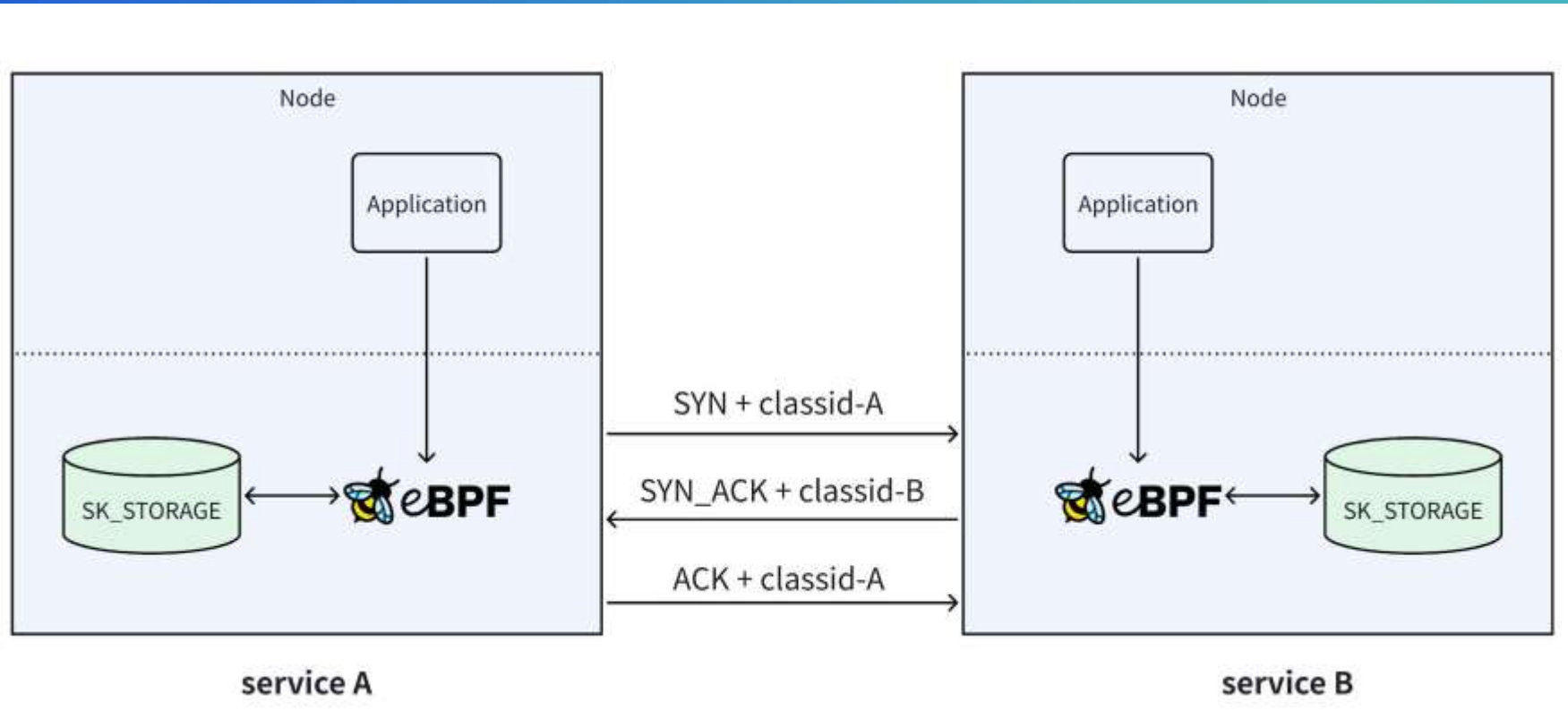
③ 实现原理-框架

- 配置下发至eBPF maps
- bpf prog hook tc
- 数据通过perf buffer上送
  - 流量计数信息
  - TCP异常信息



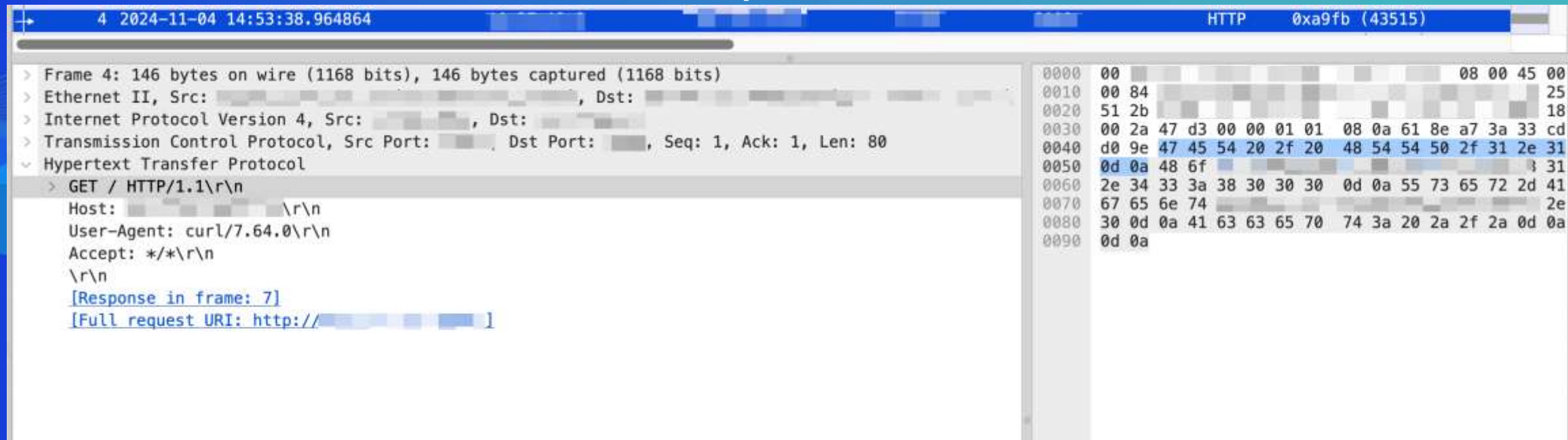
③ 实现原理-TCP option

- 数据包携带TCP option
- sk_storage存储
  - 提升性能
- 过滤项
  - 五元组
  - 源classid
  - 目的classid
  - 7层协议



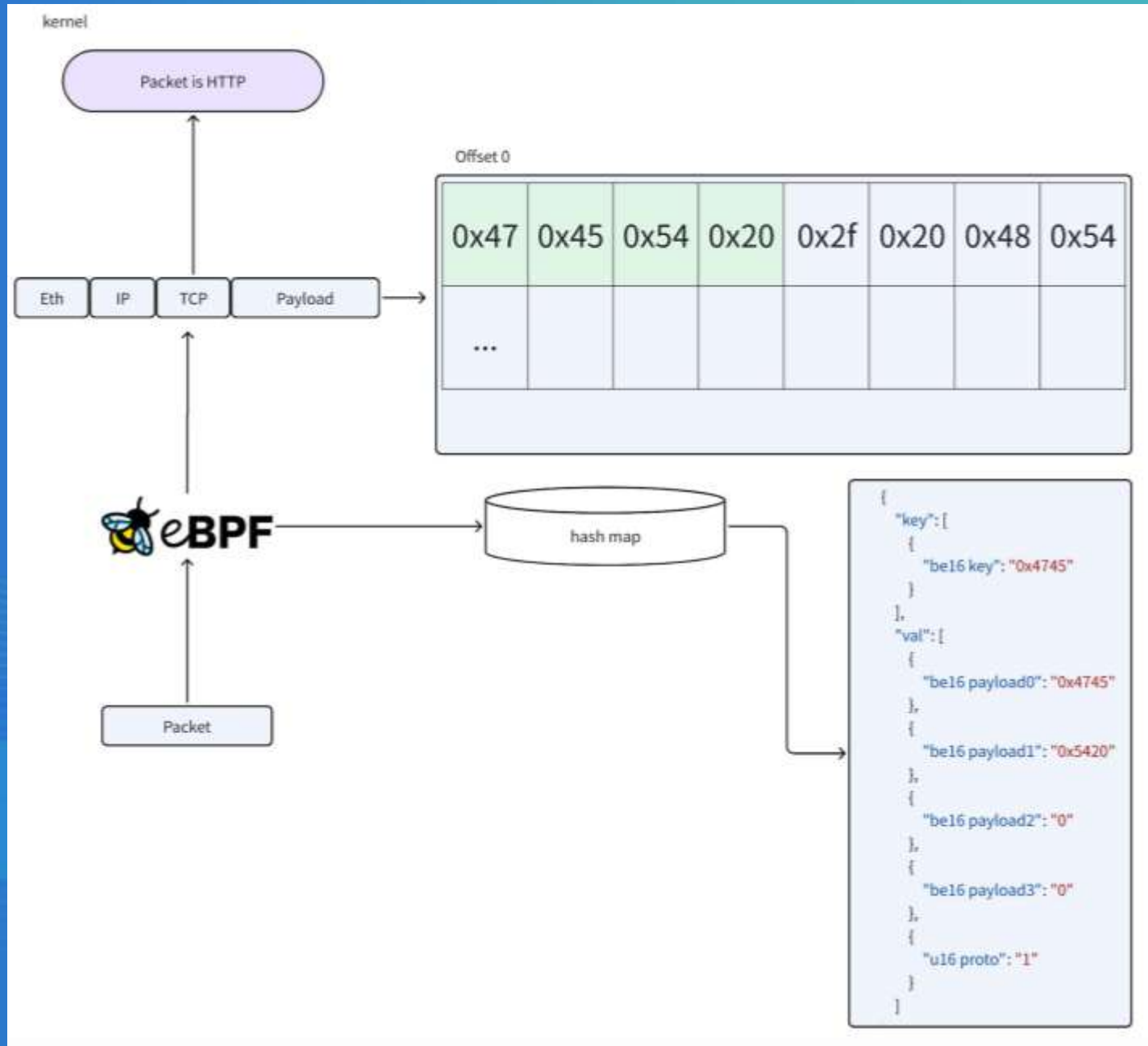
③ 实现原理-7层协议推断

- 常规的7层协议推断方法
  - 易出错
  - 不通用
  - if (buf[0] == ‘G’ && buf[1] == ‘E’ && buf[2] == ‘T’)



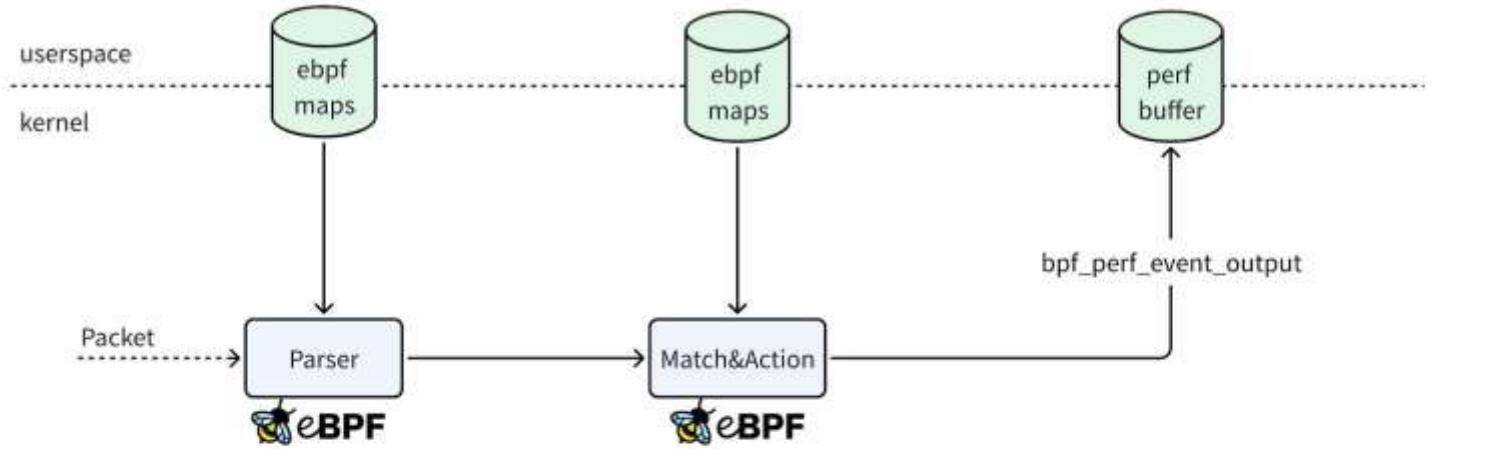
③ 实现原理-7层匹配

- 通用的7层协议推断
  - hash map存储特征
  - if offset key in hash map
  - if (payload & val == val)



③ 实现原理-示例

- 规则示例
  - match
  - action



Match

```
"dst": [
  "0.0.0.0/0"
],
"proto": [
  "tcp"
],
"proto7": [
  "http"
],
"dport": [
  "80"
],
"dst_cgroup": [
  "2001120"
```

Action

```
"action": "ACCEPT",
"action_option": {
  "CAPTURE": "id=1,interval=1,payload=40",
  "STAT": "id=1,bytes,pkts"
}
```

③ 实现原理-性能影响

- socketperf
  - 损耗低

版本	msg/sec	msg/sec相对于基线影响
基线	3030345	0%
采集&可观测	2947298	-2.74%

④

展望

④ 展望

- 海量数据处理和性能优化
- 丰富根因分析、智能检测等AI能力
- 深入探索Tracing+Continuous Profiling相结合
- 自动化抓包
- 7层域名分析
- 丢包日志

⑤ Q&A

Q&A

欢迎关注【字节跳动SYS Tech】公众号

我们是字节跳动 STE 团队 (System Technologies&Engineering，系统技术与工程）， 聚焦系统技术领域的前沿技术动态，技术创新与实践、行业技术热点等，期待与你的交流。

第三届 eBPF开发者大会

www.ebpftravel.com

⑥ Thank You!

中国·西安