GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

「CNCF云原生」专场

本期议题：eBPF在旷视PaaS平台的落地实践

王续 2021年07月10日

目录

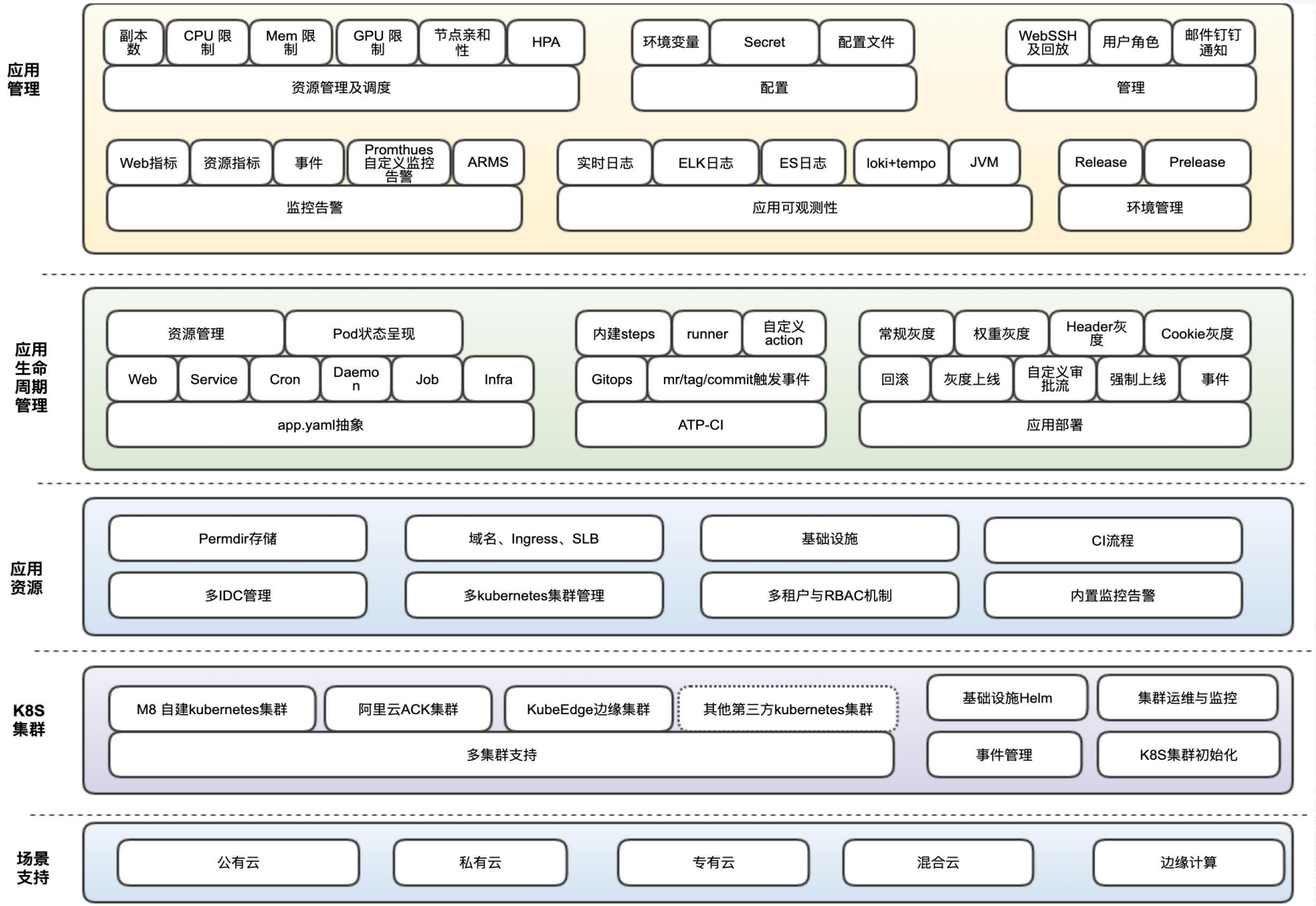
愿景：

面向全体研发，在**混合云**场景下，定义基于Kubernetes的**Application**概念，提供CI和周边基础设施，形成内部完善的、面向**云原生**的、**GitOps**驱动的**PaaS**平台

现状：

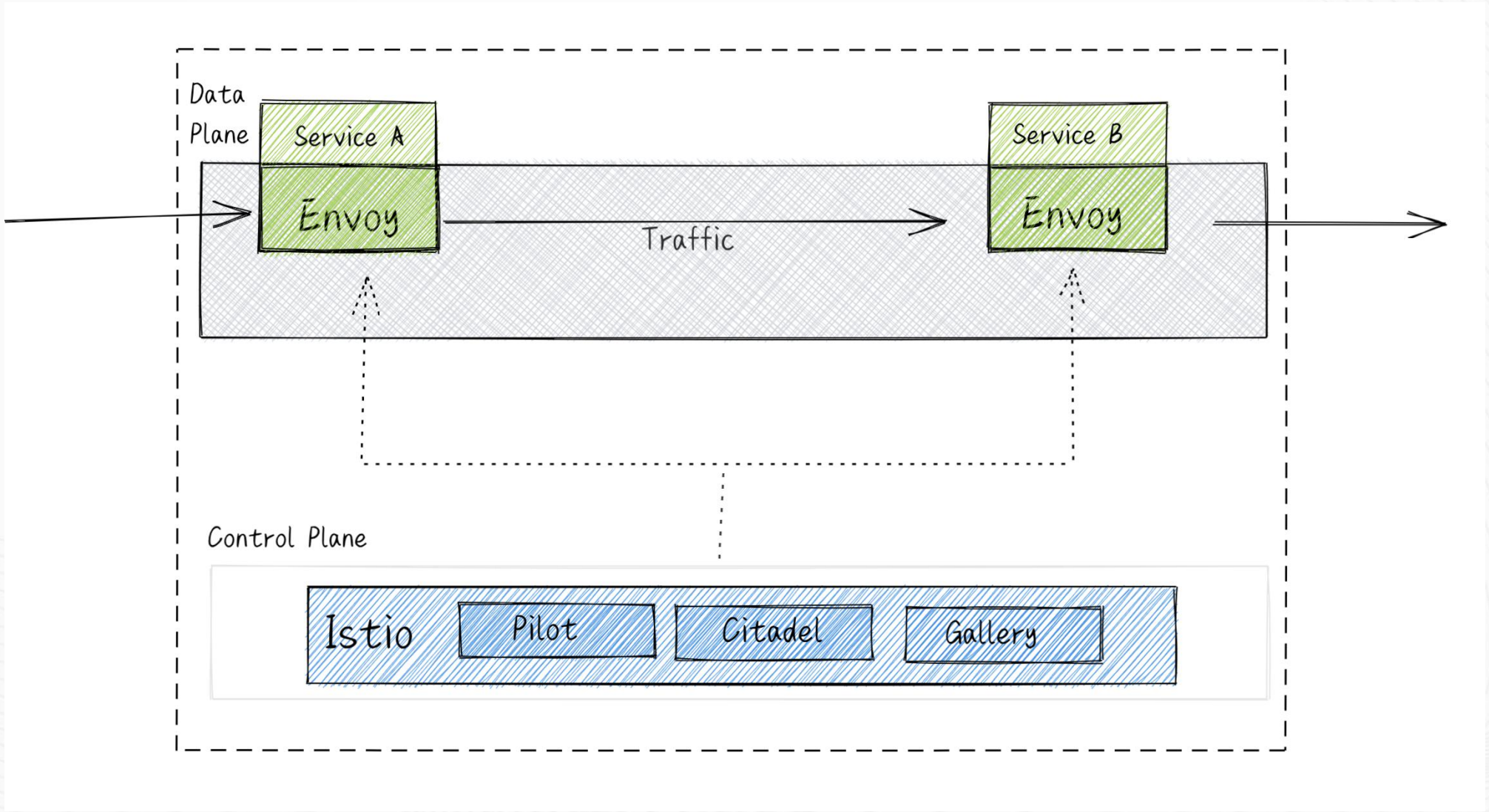
900+个应用，**10+**个集群，每日**200+**次上线

架构


应用管理
副本数
CPU 限制
Mem 限制
GPU 限制
节点亲和性
HPA
资源管理及调度
环境变量
Secret
配置文件
配置
WebSSH及回放
用户角色
邮件钉钉通知
管理
Web指标
资源指标
事件
Promthues自定义监控告警
ARMS
监控告警
实时日志
ELK日志
ES日志
loki+tempo
JVM
应用可观测性
Release
Prelease
环境管理
应用生命周期管理
资源管理
Pod状态呈现
Web
Service
Cron
Daemon
Job
Infra
app.yaml抽象
内建steps
runner
自定义action
Gitops
mr/tag/commit触发事件
ATP-CI
常规灰度
权重灰度
Header灰度
Cookie灰度
回滚
灰度上线
自定义审批流
强制上线
事件
应用部署
应用资源
Permdir存储
域名、Ingress、SLB
基础设施
CI流程
多IDC管理
多kubernetes集群管理
多租户与RBAC机制
内置监控告警
K8S集群
M8 自建kubernetes集群
阿里云ACK集群
KubeEdge边缘集群
其他第三方kubernetes集群
多集群支持
基础设施Helm
集群运维与监控
事件管理
K8S集群初始化
场景支持
公有云
私有云
专有云
混合云
边缘计算


调试
观测
高性能

一种解法


Data
Plane
Service A
Envoy
Service B
Envoy
Traffic
Control Plane
Istio
Pilot
Citadel
Gallery

再优雅一点?

GOTC

应用无感

高性能

技术成本低

The BSD Packet Filter: A New Architecture for User-level Packet Capture*

Steven McCanne† and Van Jacobson†
Lawrence Berkeley Laboratory
One Cyclotron Road
Berkeley, CA 94720
mccanne@ee.lbl.gov, van@ee.lbl.gov

December 19, 1992

Abstract

Many versions of Unix provide facilities for user-level packet capture, making possible the use of general purpose workstations for network monitoring. Because network monitors run as user-level processes, packets must be copied across the kernel/user-space protection boundary. This copying can be minimized by deploying a kernel agent called a *packet filter*, which discards unwanted packets as early as possible. The original Unix packet filter was designed around a stack-based SunOS, the Ultrix Packet Filter[2] in DEC's Ultrix and Snoop in SGI's IRIX.

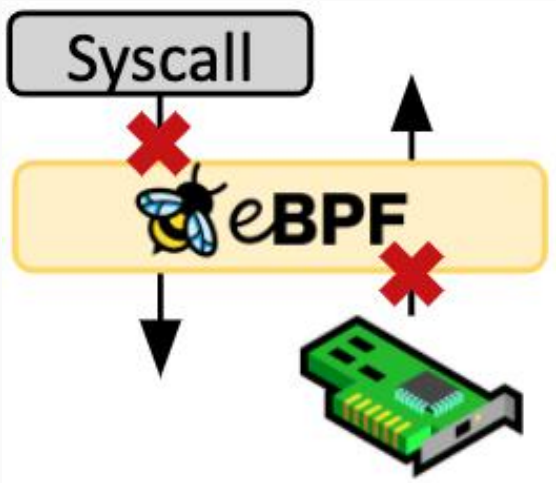
These kernel facilities derive from pioneering work done at CMU and Stanford to adapt the Xerox Alto 'packet filter' to a Unix kernel[8]. When completed in 1980, the CMU/Stanford Packet Filter, CSPF, provided a much needed and widely used facility. However on today's machines its performance, and the performance of its descendents, leave much to be desired — a design that was entirely appropriate for a 64KB PDP-11 is simply not a good match to a 16MB Sparcstation

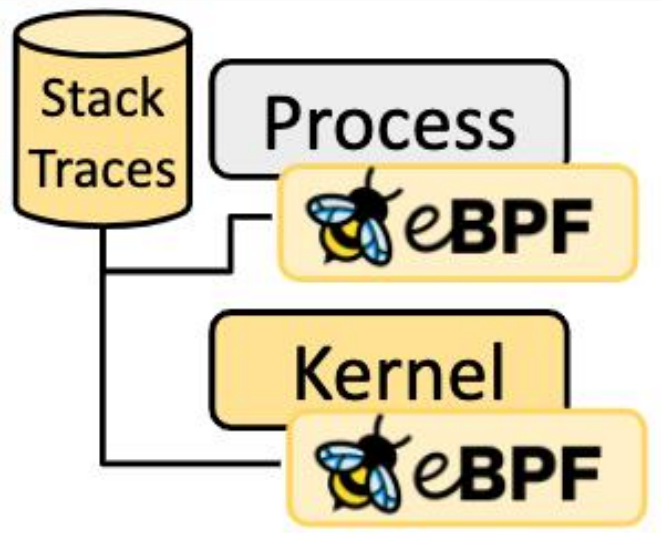
BPF & eBPF

GOTC

维度	cBPF	eBPF
内核版本	Linux 2.1.75（1997年）	Linux 3.18（2014年）[4.x for kprobe/uprobe/tracepoint/perf-event]
寄存器数目	2个： A, X	10个： R0–R9, 另外 R10 是一个只读的帧指针
寄存器宽度	32位	64位
存储	16 个内存位: M[0–15]	512 字节堆栈，无限制大小的 “map” 存储
限制的内核调用	非常有限，仅限于 JIT 特定	有限，通过 bpf_call 指令调用
目标事件	数据包、 seccomp-BPF	数据包、内核函数、用户函数、跟踪点 PMCs 等

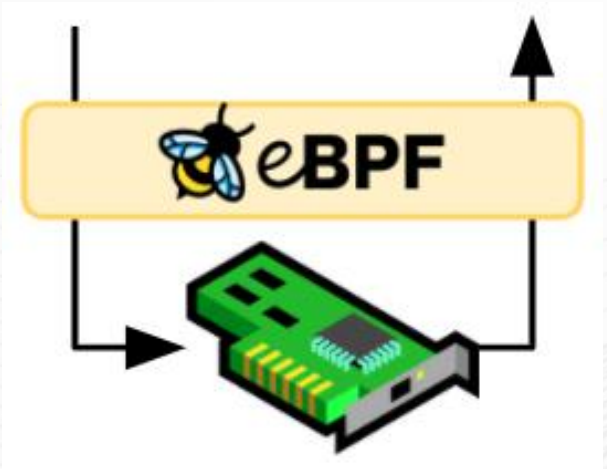
初识eBPF

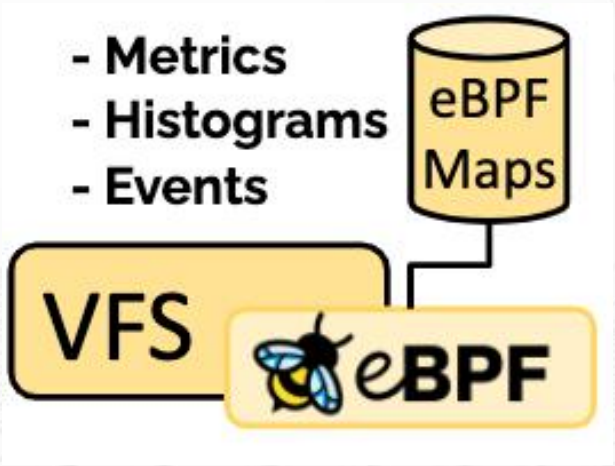




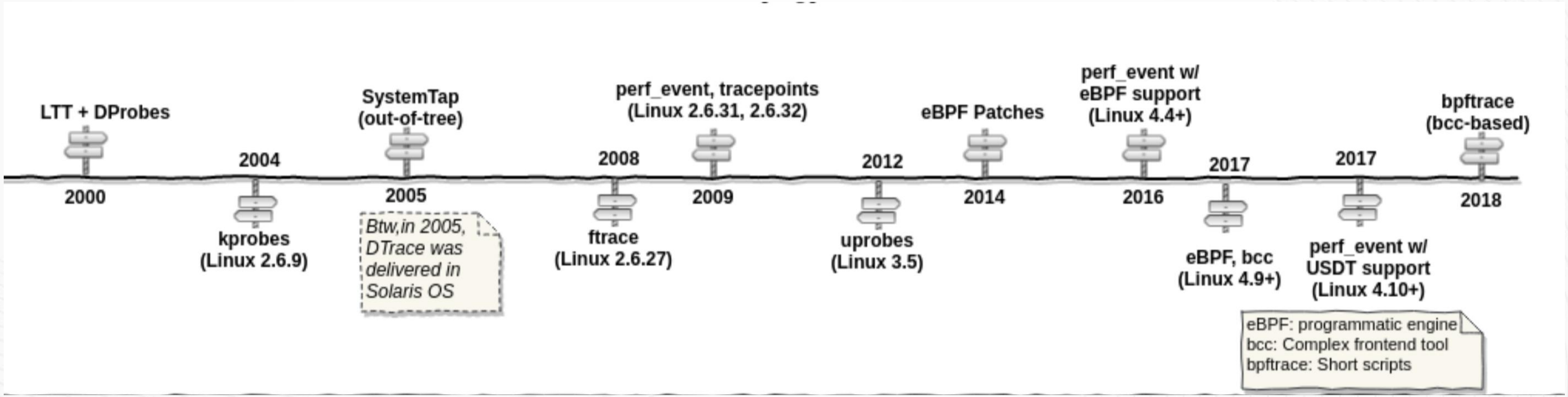
安全　性能分析

网络　观测&监控

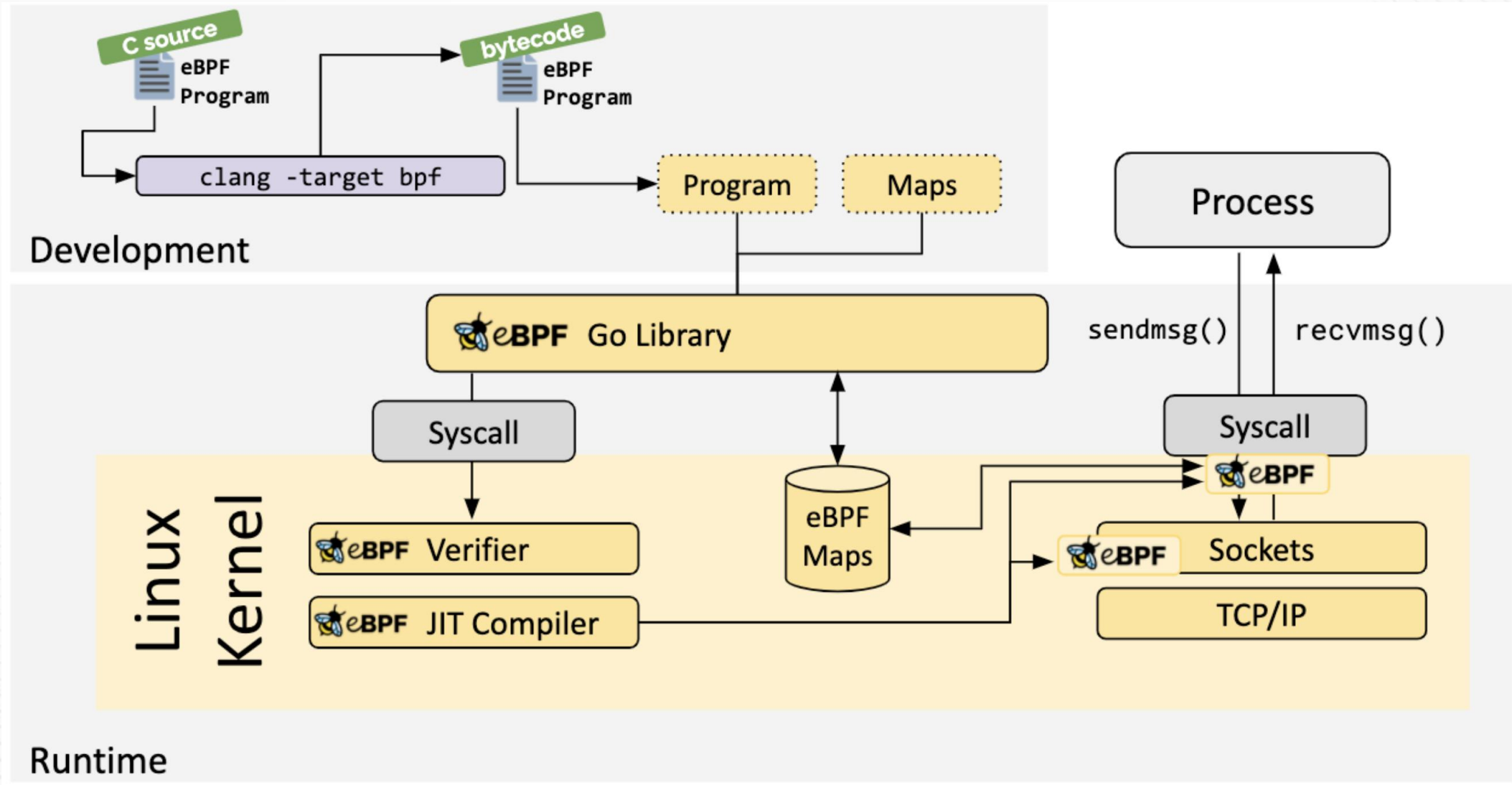




eBPF历史

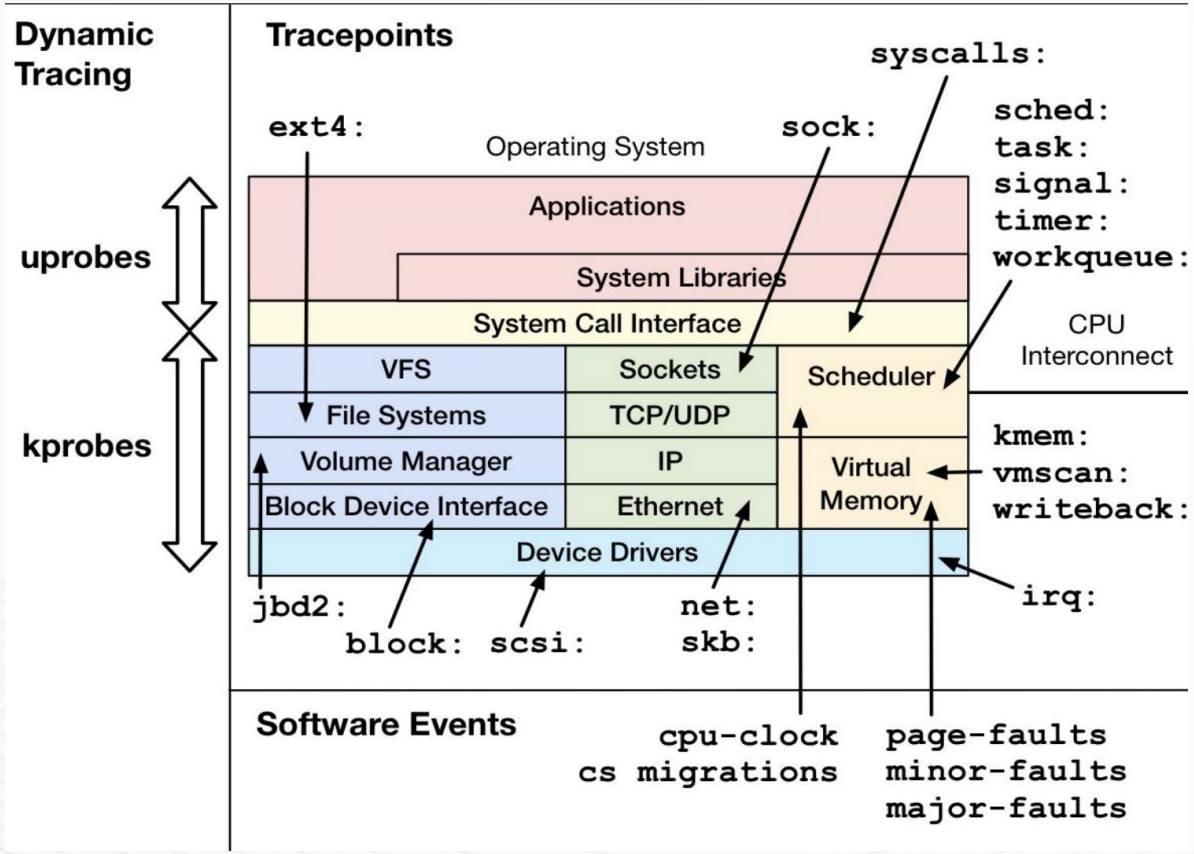

LTT + DProbes
2000
2004
kprobes
(Linux 2.6.9)
SystemTap
(out-of-tree)
2005
Btw,in 2005,
DTrace was
delivered in
Solaris OS
2008
ftrace
(Linux 2.6.27)
perf_event, tracepoints
(Linux 2.6.31, 2.6.32)
2009
2012
uprobes
(Linux 3.5)
eBPF Patches
2014
perf_event w/
eBPF support
(Linux 4.4+)
2016
2017
eBPF, bcc
(Linux 4.9+)
2017
perf_event w/
USDT support
(Linux 4.10+)
bpftrace
(bcc-based)
2018
eBPF: programmatic engine
bcc: Complex frontend tool
bpftrace: Short scripts

eBPF加载


C source
eBPF Program
bytecode
eBPF Program
clang -target bpf
Program
Maps
Development
Process
eBPF Go Library
sendmsg()
recvmsg()
Syscall
Syscall
eBPF
eBPF Maps
eBPF Verifier
eBPF
eBPF Sockets
eBPF JIT Compiler
TCP/IP
Linux Kernel
Runtime

```
# ./tcplife.bt
PID   COMM       LADDR           LPORT RADDR           RPORT TX_KB RX_KB MS
20976 ssh        127.0.0.1       56766 127.0.0.1       22          6 10584 3059
20977 sshd       127.0.0.1       22    127.0.0.1       56766   10584     6 3059
14519 monitord   127.0.0.1       44832 127.0.0.1       44444       0     0 0
4496  Chrome_IOT 7f00:6:5ea7::a00:0 42846 0:0:bb01::       443         0     3 12441
4496  Chrome_IOT 7f00:6:5aa7::a00:0 42842 0:0:bb01::       443         0     3 12436
4496  Chrome_IOT 7f00:6:62a7::a00:0 42850 0:0:bb01::       443         0     3 12436
4496  Chrome_IOT 7f00:6:5ca7::a00:0 42844 0:0:bb01::       443         0     3 12442
4496  Chrome_IOT 7f00:6:60a7::a00:0 42848 0:0:bb01::       443         0     3 12436
4496  Chrome_IOT 10.0.0.65       33342 54.241.2.241    443         0     3 10717
4496  Chrome_IOT 10.0.0.65       33350 54.241.2.241    443         0     3 10711
4496  Chrome_IOT 10.0.0.65       33352 54.241.2.241    443         0     3 10712
14519 monitord   127.0.0.1       44832 127.0.0.1       44444       0     0 0
```

源码分析



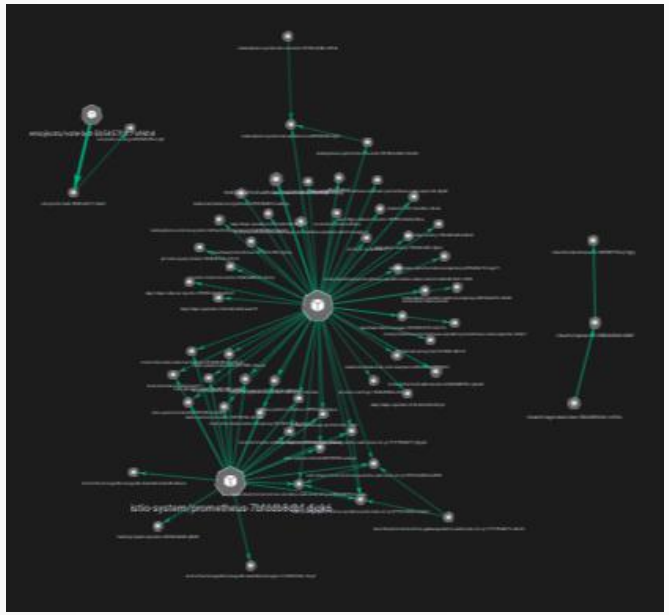

```
// session ended: calculate lifespan and print
if ($newstate == TCP_CLOSE && @birth[$sk]) {
    $delta_ms = (nsecs - @birth[$sk]) / 1e6;
    $lport = $sk->__sk_common.skc_num;
    $dport = $sk->__sk_common.skc_dport;
    $dport = ($dport >> 8) | (($dport << 8) & 0xff00);
    $tp = (struct tcp_sock *)$sk;
    $pid = @skpid[$sk];
    $comm = @skcomm[$sk];
    if ($comm == "") {
        // not cached, use current task
        $pid = pid;
        $comm = comm;
    }

    $family = $sk->__sk_common.skc_family;
    $saddr = ntop(0);
    $daddr = ntop(0);
    if ($family == AF_INET) {
        $saddr = ntop(AF_INET, $sk->__sk_common.skc_rcv_saddr);
        $daddr = ntop(AF_INET, $sk->__sk_common.skc_daddr);
    } else {
        // AF_INET6
        $saddr = ntop(AF_INET6,
            $sk->__sk_common.skc_v6_rcv_saddr.in6_u.u6_addr8);
        $daddr = ntop(AF_INET6,
            $sk->__sk_common.skc_v6_daddr.in6_u.u6_addr8);
    }
    printf("%-5d %-10.10s %-15s %-5d %-15s %-6d ", $pid,
        $comm, $saddr, $lport, $daddr, $dport);
    printf("%5d %5d %d\n", $tp->bytes_acked / 1024,
        $tp->bytes_received / 1024, $delta_ms);

    delete(@birth[$sk]);
    delete(@skpid[$sk]);
    delete(@skcomm[$sk]);
```

Pixie

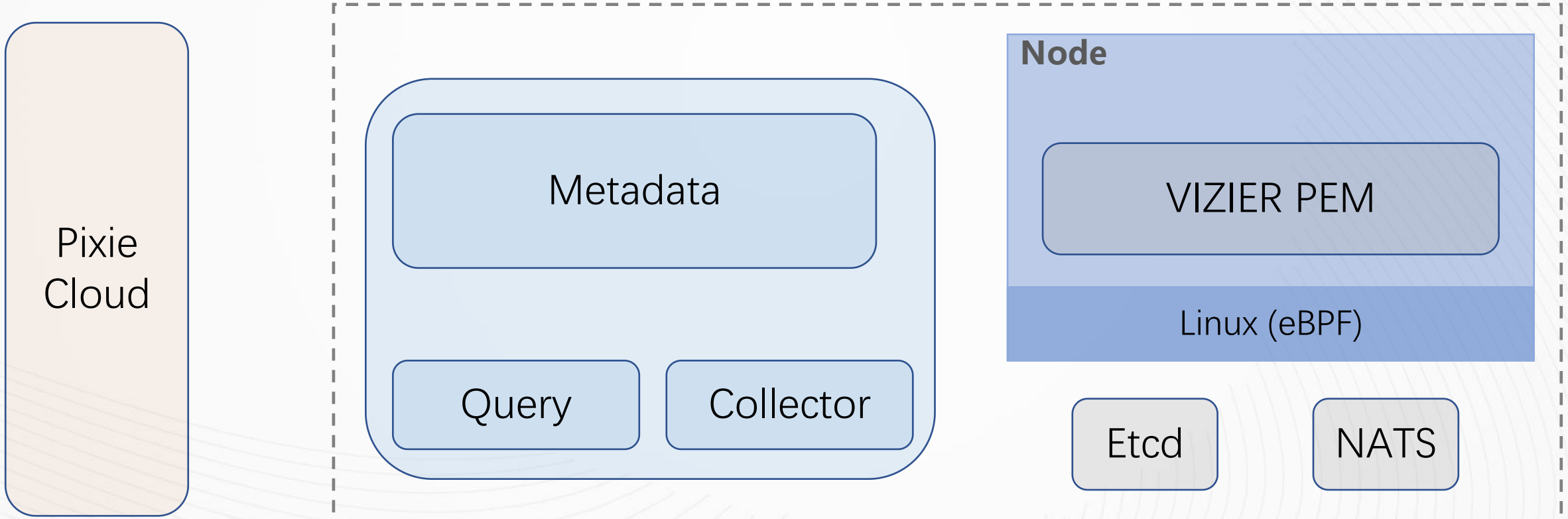
GOTC



PID	PROCESS NAME	ASID	TIMES...	VI...	AVERAG...
22,953	/usr/bin/promtail -config.file=/etc/promtail/promtail.yaml -client.url=http://loki-distributo...	9	2021/6/29...	6.8 GB	414.1 MB
22,953	/usr/bin/promtail -config.file=/etc/promtail/promtail.yaml -client.url=http://loki-distributo...	9	2021/6/29...	6.8 GB	414.1 MB
23,951	haproxy -W -db -f /usr/local/etc/haproxy/haproxy.cfg	9	2021/6/29...	3 GB	84.8 MB
23,951	haproxy -W -db -f /usr/local/etc/haproxy/haproxy.cfg	9	2021/6/29...	3 GB	84.8 MB
23,309	/app/mcd-admission	9	2021/6/29...	3 GB	28.3 MB
23,309	/app/mcd-admission	9	2021/6/29...	3 GB	28.3 MB
22,305	/app/mcd-admission	9	2021/6/29...	2.9 GB	28 MB
22,305	/app/mcd-admission	9	2021/6/29...	2.9 GB	28 MB
6,983	/app/mcd-admission	9	2021/6/29...	2.8 GB	27.8 MB
6,983	/app/mcd-admission	9	2021/6/29...	2.8 GB	27.8 MB

RE...	RE...	RE...	REQ_PATH	RESP_STATUS	RESP_BODY	LA...	SERVICE
10.0.1....	40040	POST	/_bulk	200	{ took: 81, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	82.3 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{ took: 72, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	73.3 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{ took: 67, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	68.1 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{ took: 94, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	95.4 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{ took: 52, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	53.3 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{ took: 62, errors: false, items: [{ index: { _index: queue_metrics_d1bd92a3b039400cbafc60...	62.9 ms	clearml/ela...
10.0.1....	40040	POST	/_bulk	200	{"took":40,"errors":false,"items":[{"index":{"_index":"worker_stats_d1bd92a3b039400cbafc60a7a5b1e52b_2021-0...	40.7 ms	clearml/ela...

架构


Pixie
Cloud
Metadata
Query
Collector
Node
VIZIER PEM
Linux (eBPF)
Etcd
NATS

Pixie

GOTC

Table

TI…	UPID	RE…	RE…	M…	REQ_HEADERS	RE…	REQ_PATH	RE…
4/20/2…	00000032-0000-325…	192.16…	47082	1	{ Connection: cl…	GET	/catalogue?page=1&size=…	

```
  },
 resp_status: 500,
 resp_message: Internal Server Error,
 resp_body: {
   error: Do: database connection error,
   status_code: 500,
   status_text: Internal Server Error
  },
 resp_body_size: 98,
 latency: 440476,
 pod: px-sock-shop/catalogue-8f6fdb6d8-24x72,
 service: px-sock-shop/catalogue
```

Pixie

GOTC

```
def http_data(start_time: str, num_head: int):
    df = px.DataFrame(table='http_events', start_time=start_time)

    df.namespace = df.ctx['namespace']
    df.node = df.ctx['node']
    df.pod = df.ctx['pod']
    df.pid = px.upid_to_pid(df.upid)

    # Remove some columns.
    df = df.drop(['upid', 'trace_role', 'content_type', 'minor_version'])

    # Restrict number of results.
    df = df.head(num_head)

    return df
```

Pixie

```
constexpr DataElement kHTTPElements[] = {
    canonical_data_elements::kTime,
    canonical_data_elements::kUPID,
    canonical_data_elements::kRemoteAddr,
    canonical_data_elements::kRemotePort,
    canonical_data_elements::kTraceRole,
    {"minor_version", "HTTP minor version, HTTP1 uses 1, HTTP2 set this value to 0",
     types::DataType::INT64,
     types::SemanticType::ST_NONE,
     types::PatternType::GENERAL_ENUM},
    {"content_type", "Type of the HTTP payload, can be JSON or protobuf",
     types::DataType::INT64,
     types::SemanticType::ST_NONE,
     types::PatternType::GENERAL_ENUM,
     &kHTTPContentTypeDecoder},
    {"req_headers", "Request headers in JSON format",
     types::DataType::STRING,
     types::SemanticType::ST_NONE,
     types::PatternType::STRUCTURED},
    ...
};
// clang-format on

constexpr auto kHTTPTable = DataTableSchema("http_events", "HTTP request-response pair events",
                                            kHTTPElements, std::chrono::milliseconds{1000});
```

Pixie

GOTC

```
SocketTraceConnector::SocketTraceConnector(std::string_view source_name)
    : SourceConnector(source_name, kTables), conn_stats_(&conn_trackers_mgr_), uprobe_mgr_(this) {
  proc_parser_ = std::make_unique<system::ProcParser>(system::Config::GetInstance());
  InitProtocolTransferSpecs();
}

    {kProtocolHTTP, TransferSpec{FLAGS_stirling_enable_http_tracing,
                                 kHTTPTableNum,
                                 {kRoleClient, kRoleServer},
                                 TRANSFER_STREAM_PROTOCOL(http)}},
```

注册transfer

```
if (tracker->state() == ConnTracker::State::kTransferring) {
  // ProcessToRecords() parses raw events and produces messages in format that are expected by
  // table store. But those messages are not cached inside ConnTracker.
  auto result = tracker->ProcessToRecords<TProtocolTraits>();
  for (auto& msg : result) {
    AppendMessage(ctx, *tracker, std::move(msg), data_table);
  }
```

解析events

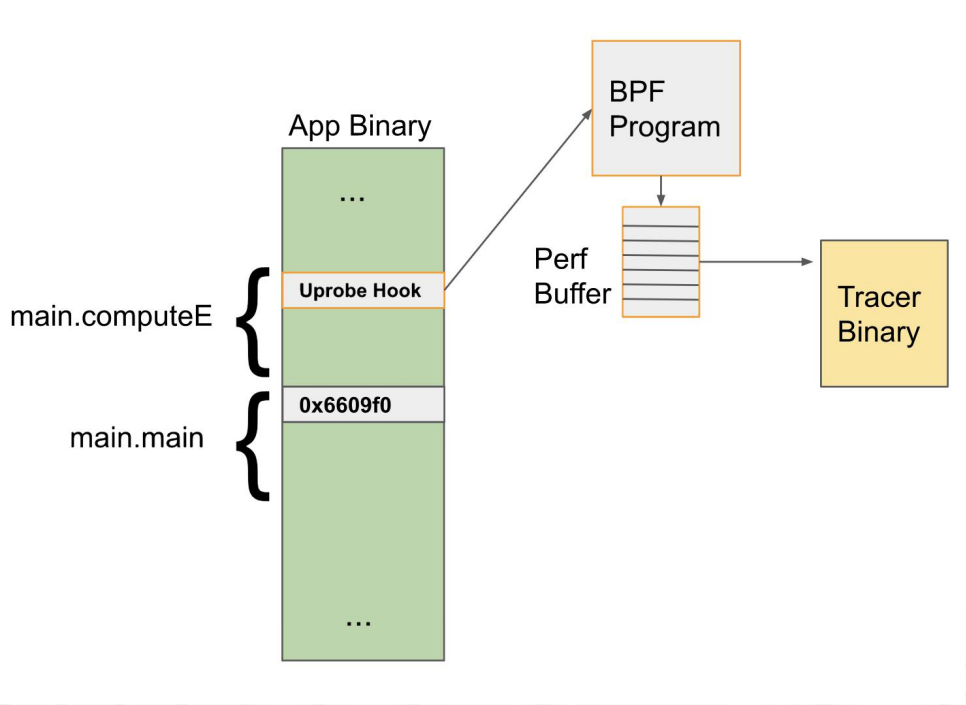
```
r.Append<r.ColIndex("remote_addr")>(conn_tracker.remote_endpoint().AddrStr());
r.Append<r.ColIndex("remote_port")>(conn_tracker.remote_endpoint().port());
r.Append<r.ColIndex("trace_role")>(conn_tracker.role());
r.Append<r.ColIndex("major_version")>(1);
r.Append<r.ColIndex("minor_version")>(resp_message.minor_version);
r.Append<r.ColIndex("content_type")>(static_cast<uint64_t>(content_type));
```

构造table

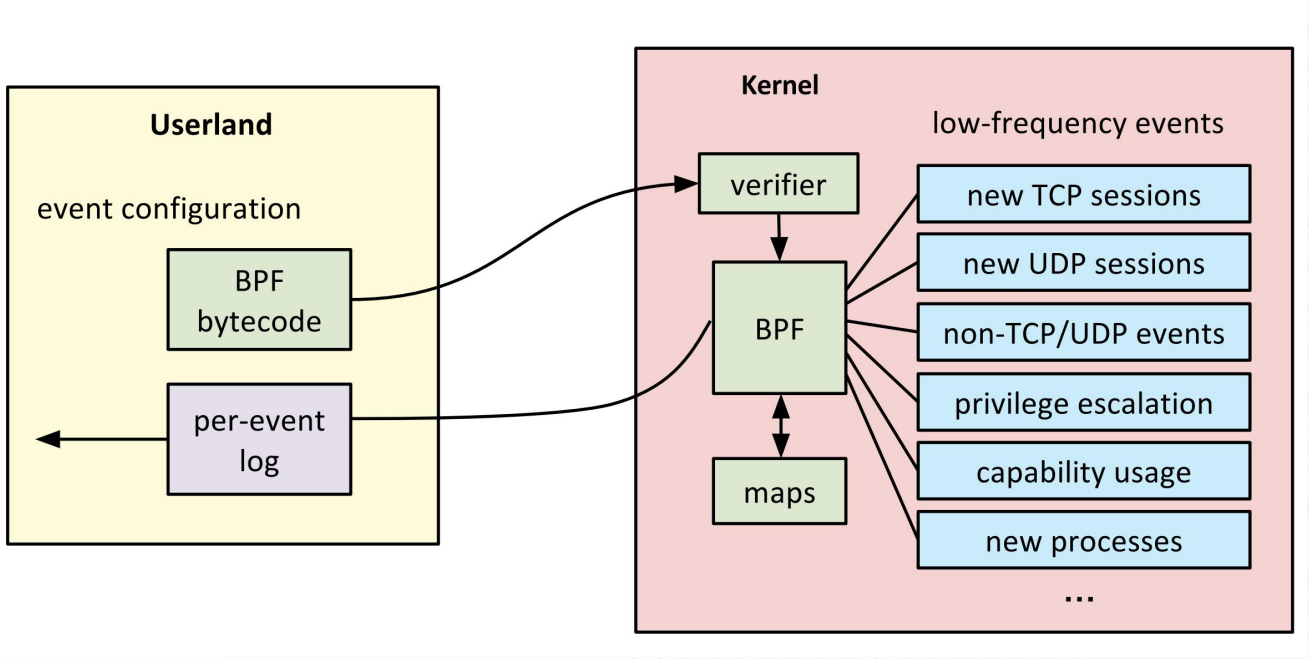
一些探索

GOTC

Remote Debug



Security



GOTC

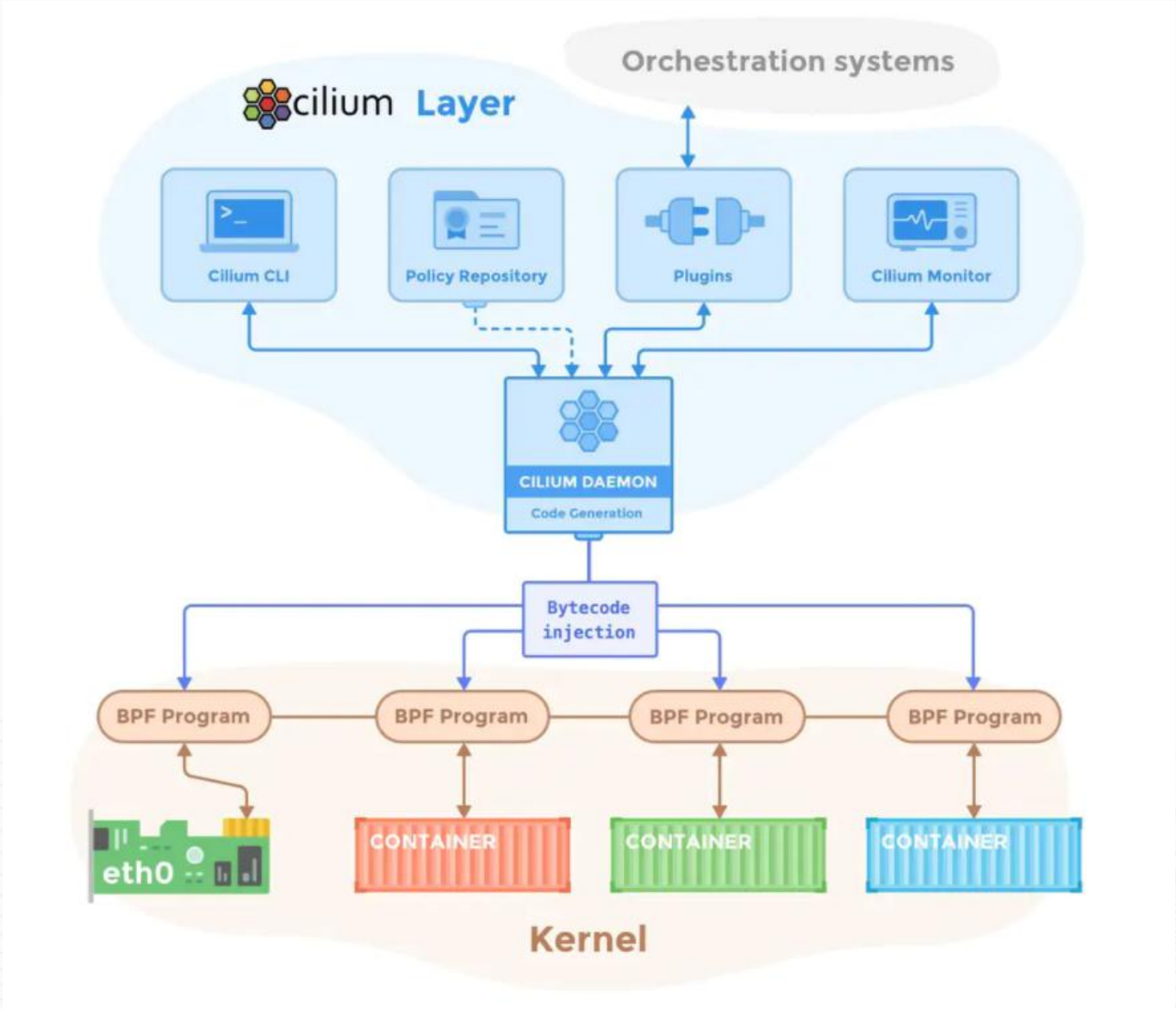
性能

Latency percentiles at 50k RPS (≈20% of maximum RPS), ms

Setup	95 %ile	99 %ile	99.5 %ile	99.99 %ile	99.999 %ile	Max Latency
IPvlan	0.6	0.8	0.9	5.7	9.6	15.8
`aws-vpc`	0.7	0.9	1	5.6	9.8	403.1
`host-gw`	0.7	0.9	1	7.4	12	202.5
`vxlan`	0.8	1.1	1.2	5.7	201.5	402.5
`--net=host`	0.5	0.7	0.7	6.4	9.9	14.8

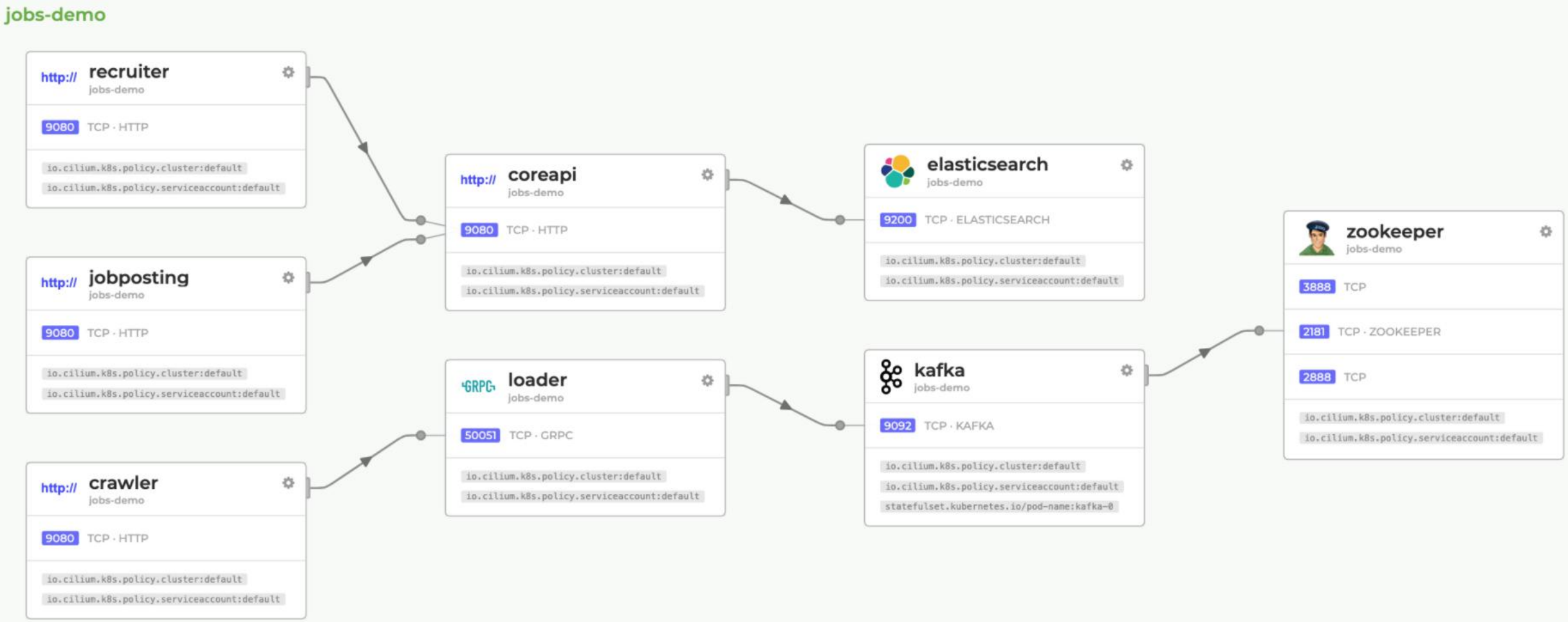
Network Policy

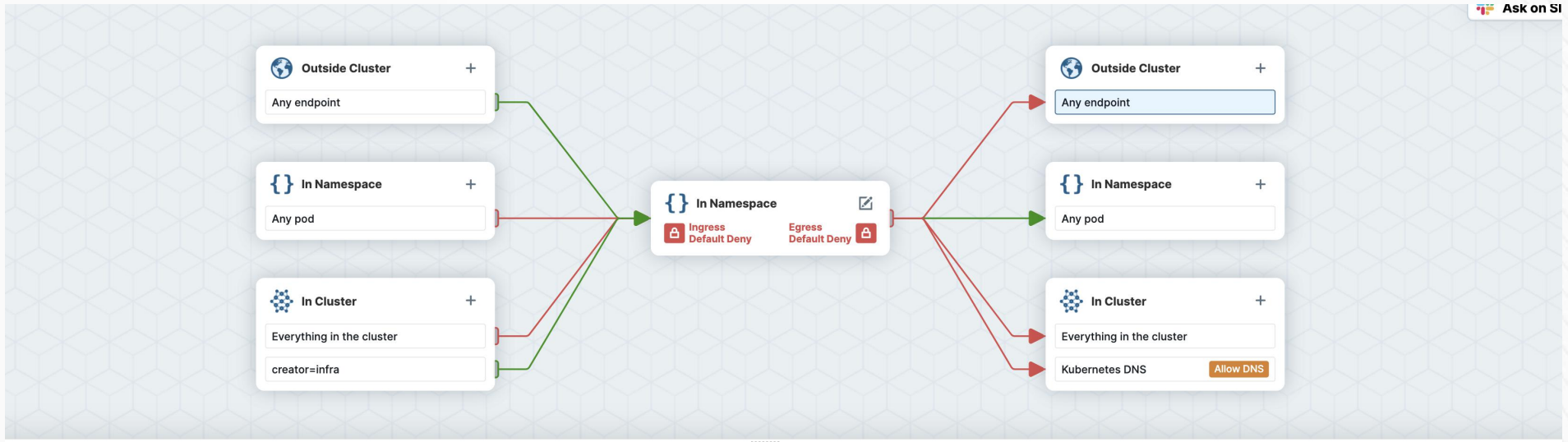
```
ingress:
- from:
  - namespaceSelector:
      matchLabels:
        user: alice
    podSelector:
      matchLabels:
        role: client
```


Orchestration systems
cilium Layer
Cilium CLI
Policy Repository
Plugins
Cilium Monitor
CILIUM DAEMON
Code Generation
Bytecode injection
BPF Program
BPF Program
BPF Program
BPF Program
eth0
CONTAINER
CONTAINER
CONTAINER
Kernel

Cilium

GOTC


jobs-demo
http:// recruiter
jobs-demo
9080 TCP · HTTP
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
http:// jobposting
jobs-demo
9080 TCP · HTTP
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
http:// crawler
jobs-demo
9080 TCP · HTTP
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
http:// coreapi
jobs-demo
9080 TCP · HTTP
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
GRPC loader
jobs-demo
50051 TCP · GRPC
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
elasticsearch
jobs-demo
9200 TCP · ELASTICSEARCH
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
kafka
jobs-demo
9092 TCP · KAFKA
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default
statefulset.kubernetes.io/pod-name:kafka-0
zookeeper
jobs-demo
3888 TCP
2181 TCP · ZOOKEEPER
2888 TCP
io.cilium.k8s.policy.cluster:default
io.cilium.k8s.policy.serviceaccount:default



Kubernetes Network Policy | Cilium Network Policy

Policy Rating | Download | Share

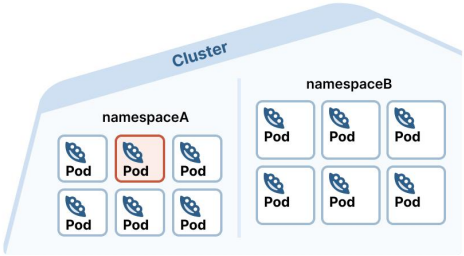
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: untitled-policy
spec:
  podSelector: {}
  ingress:
    - from:
        - ipBlock:
            cidr: 0.0.0.0/0
    - from:
        - namespaceSelector:
            matchLabels:
              creator: infra
  egress:
    - to:
        - podSelector: {}
```

Main tutorial | Flows upload

Welcome to the Network Policy Editor! *Beta*

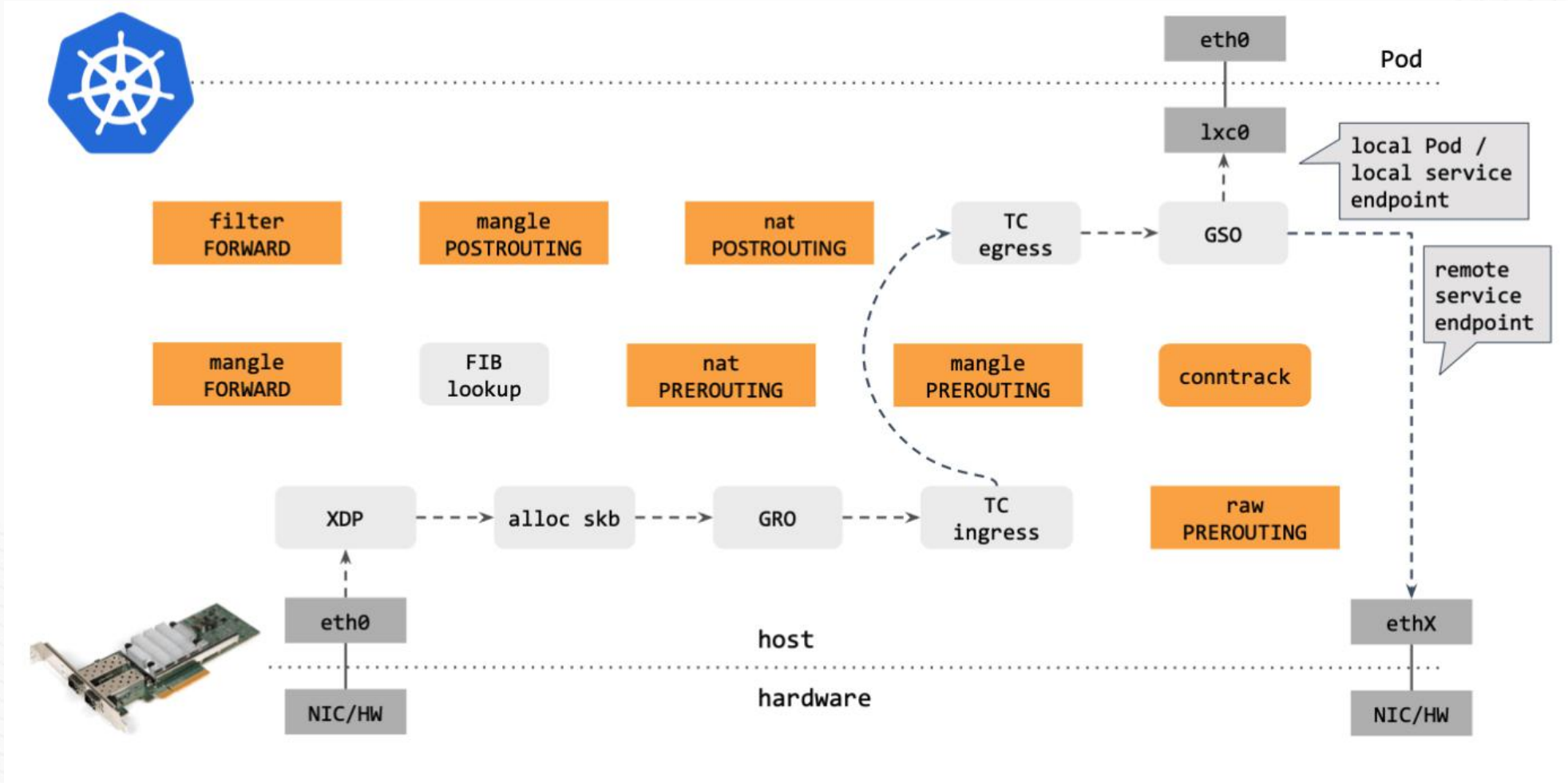
This tutorial will teach you how to create a network policy using Editor. It explains basic network policy concepts and guides yo through the steps needed to achieve the desired least- privileg security and zero-trust concepts.

Step 1. What pods do you want to secure?



First, select the pods to which the policy should be applied by

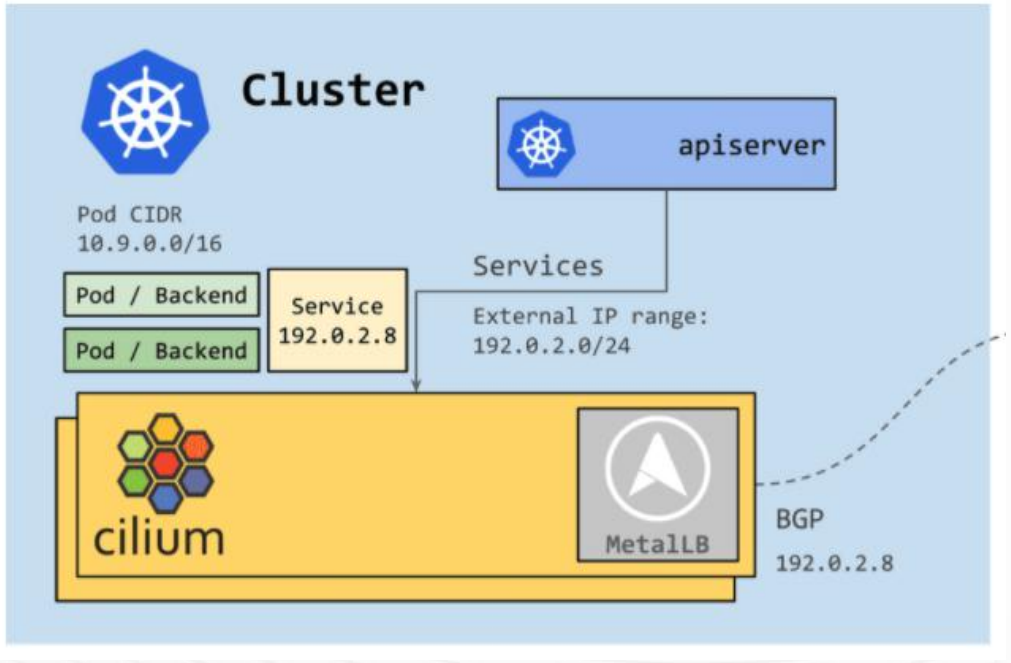
生命周期


eth0
Pod
lxc0
local Pod /
local service
endpoint
filter
FORWARD
mangle
POSTROUTING
nat
POSTROUTING
TC
egress
GSO
remote
service
endpoint
mangle
FORWARD
FIB
lookup
nat
PREROUTING
mangle
PREROUTING
conntrack
XDP
alloc skb
GRO
TC
ingress
raw
PREROUTING
eth0
ethX
host
hardware
NIC/HW
NIC/HW

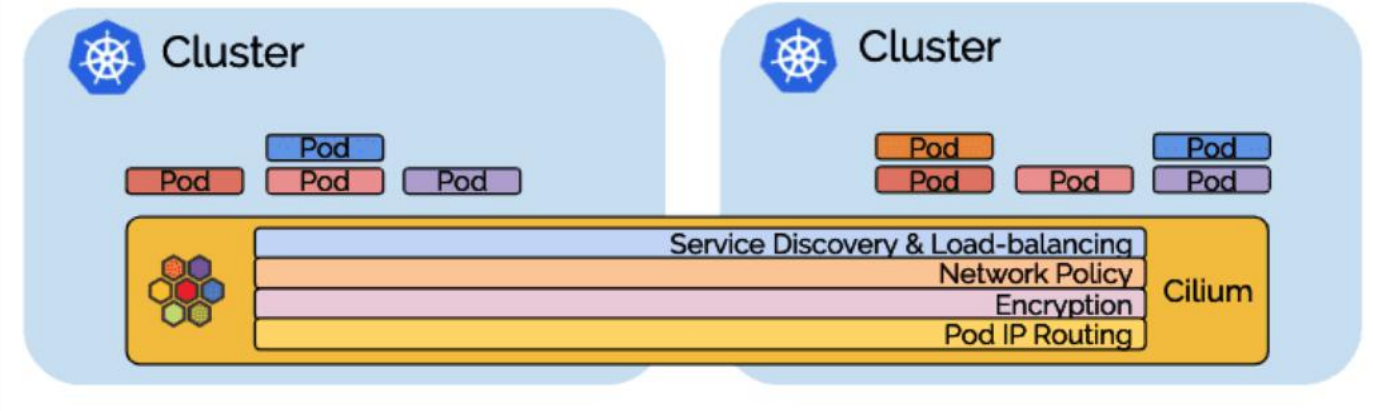
一些探索

GOTC

L4 LB



ClusterMesh



参考资料

https://github.com/icopy-site/awesome-cn/blob/master/docs/awesome/awesome-ebpf.md
http://machinezone.github.io/research/networking-solutions-for-kubernetes/
https://github.com/iovisor/bpf-docs/blob/master/Express_Data_Path.pdf
https://cilium.io/blog/2020/11/10/ebpf-future-of-networking/
https://www.kernel.org/doc/html/latest/networking/filter.html
https://www.tcpdump.org/papers/bpf-usenix93.pdf
https://github.com/sbueringer/kubecon-slides
https://developer.aliyun.com/article/779357
https://nakryiko.com/posts/bpf-ringbuf/
https://github.com/weaveworks/scope
https://github.com/pixie-labs/pixie
https://lwn.net/Articles/132196/
https://github.com/cilium/cilium
https://github.com/draios/sysdig
https://github.com/iovisor/bcc
https://ebpf.io

GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE