第三届 eBPF开发者大会
www.ebpftravel.com

基于 eBPF 的 VxLAN 性能加速及数据库应用

- ONCache：将cache引入容器overlay网络

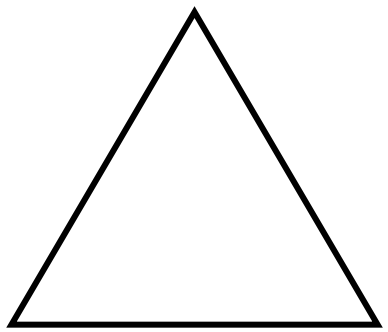
林圣凯
上海交通大学

李阳
上海星环科技有限公司

中国·西安

研究背景

- 现有容器网络技术均未兼顾兼容性、灵活性与高性能/低开销等特性。

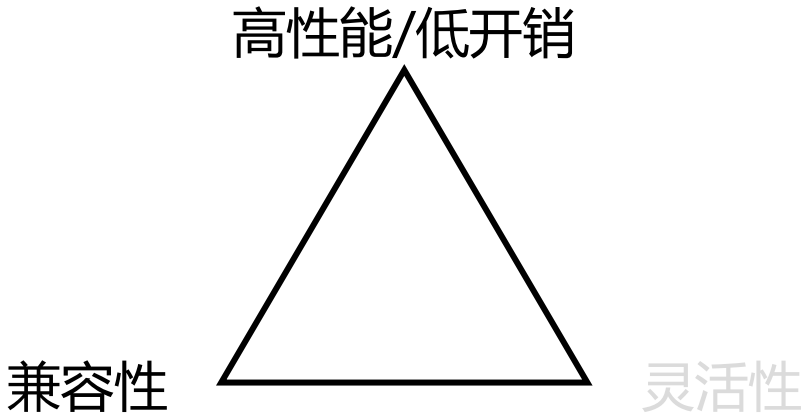


内核中报文的处理开销低；报文处理速度快

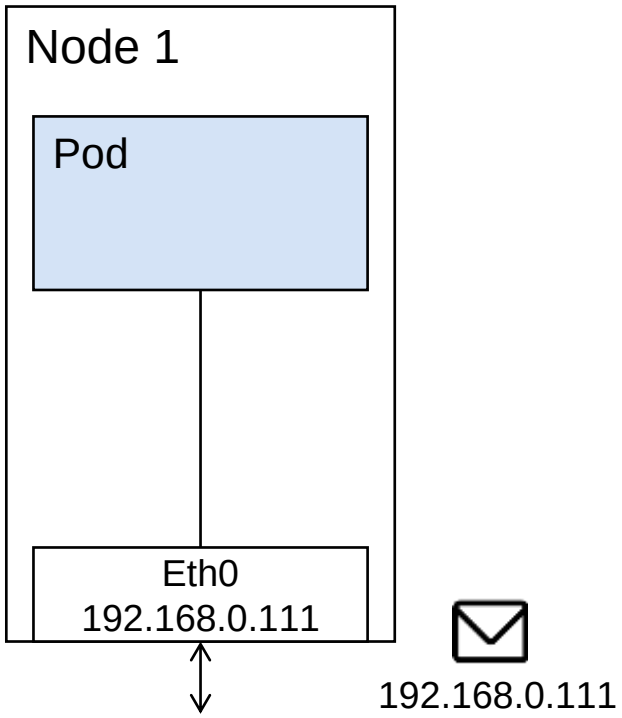
支持各种通信协议

应用开发与网络运维解耦合

现有工作：兼容性+高性能/低开销

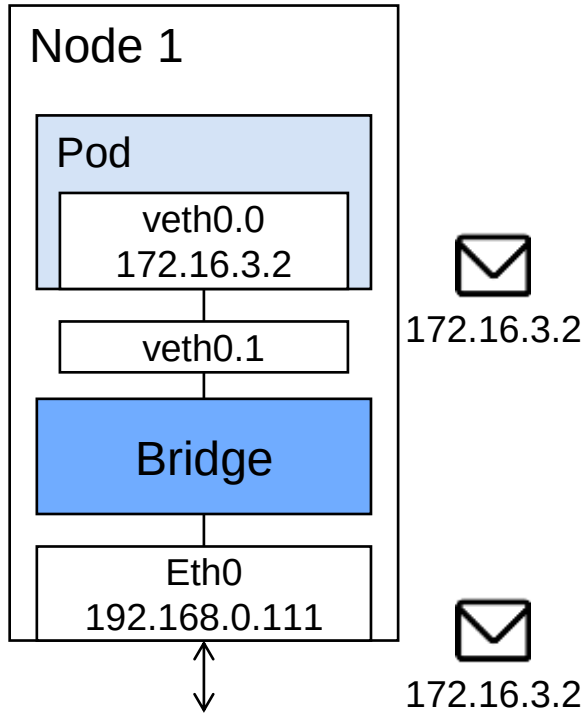


- 容器应用与物理网络耦合
- 限制了容器开发、部署的灵活性



Host network

- 容器使用主机IP
- 容器应用需协商端口号

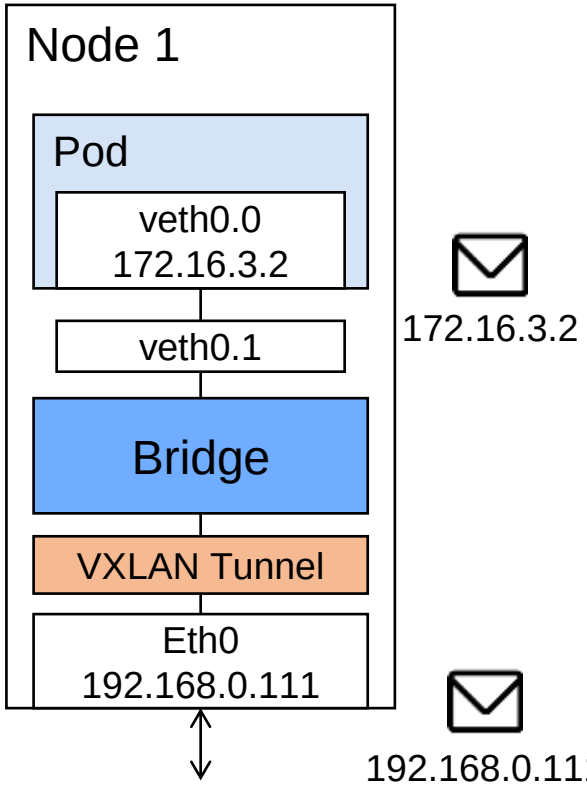


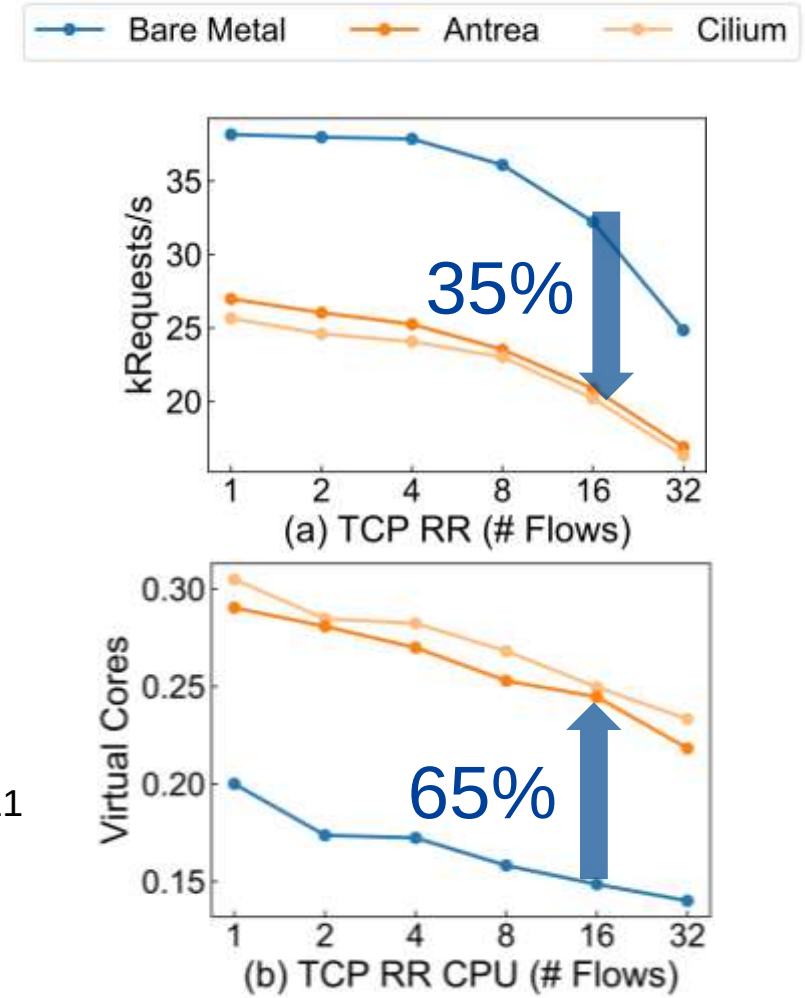
Bridge network

- 容器独立IP
- 物理网络负责路由容器IP

现有工作：兼容性+灵活性

- 容器应用与物理网络之间解耦合
- 隧道引入了显著开销

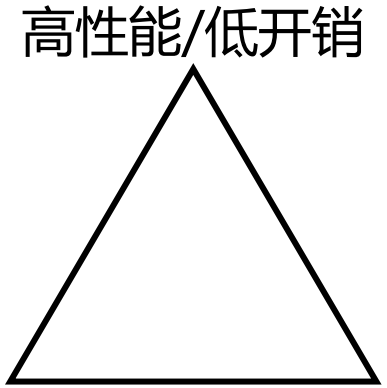




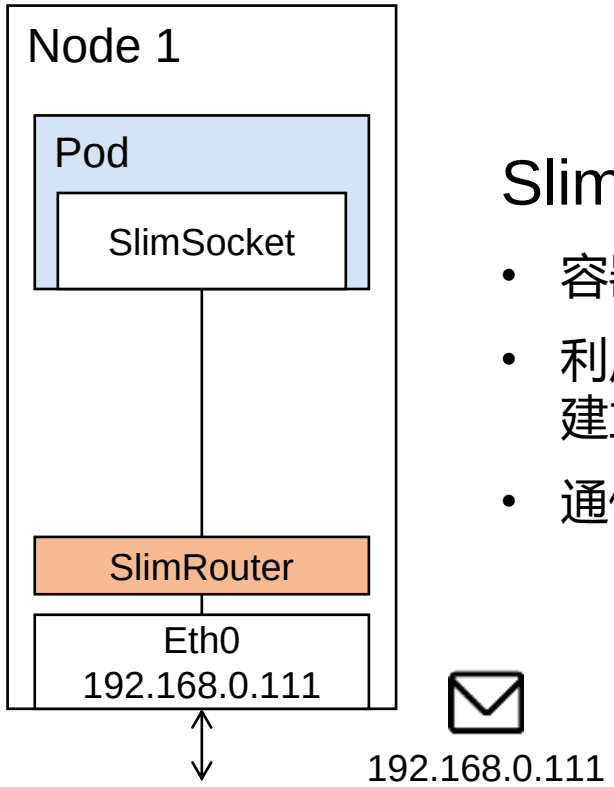
Overlay network

- 容器独立IP
- 使用VXLAN等隧道技术二次封装报文

现有工作：灵活性+高性能/低开销



- 容器应用与物理网络之间解耦合
- 使用host socket取代隧道，不支持UDP，ICMP等协议
- 难以实际落地



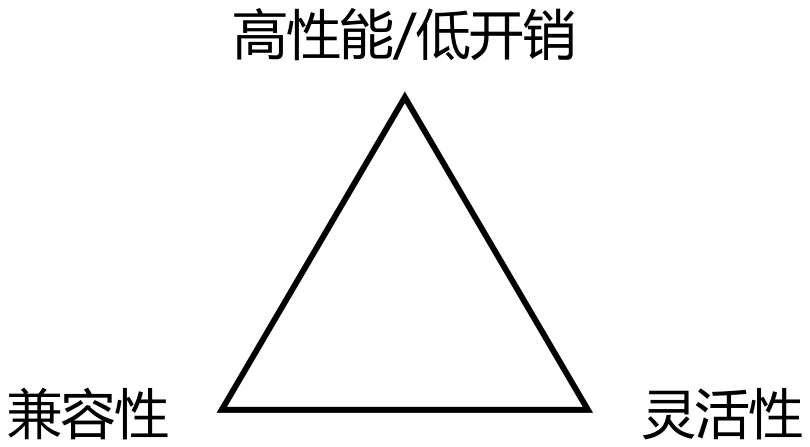
Slim [1]

- 容器自有IP
- 利用中间层使容器仍可用自身IP建立连接
- 通信时容器实际直接用host网卡

[1] Zhuo, Danyang, et al. "Slim:{OS} Kernel Support for a {Low-Overhead} Container Overlay Network." *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 2019.

ONCache: A Cache-Based Low-Overhead Container Overlay Network

Shengkai Lin[1], Shizhen Zhao[1], Peirui Cao[1], Xinchi Han[1],
Quan Tian[2], Wenfeng Liu[2], Qi Wu[2], Donghai Han[2], and Xinbing Wang[1]
[1]*Shanghai Jiao Tong University,* [2]*Broadcom*



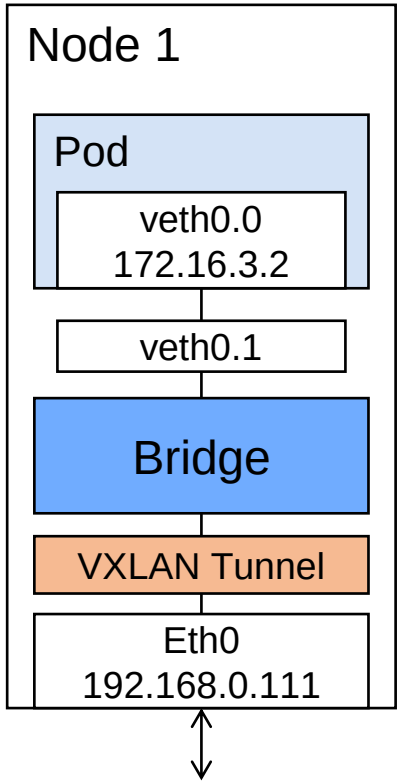
nsdi'25
22nd USENIX Symposium on Networked Systems Design and Implementation

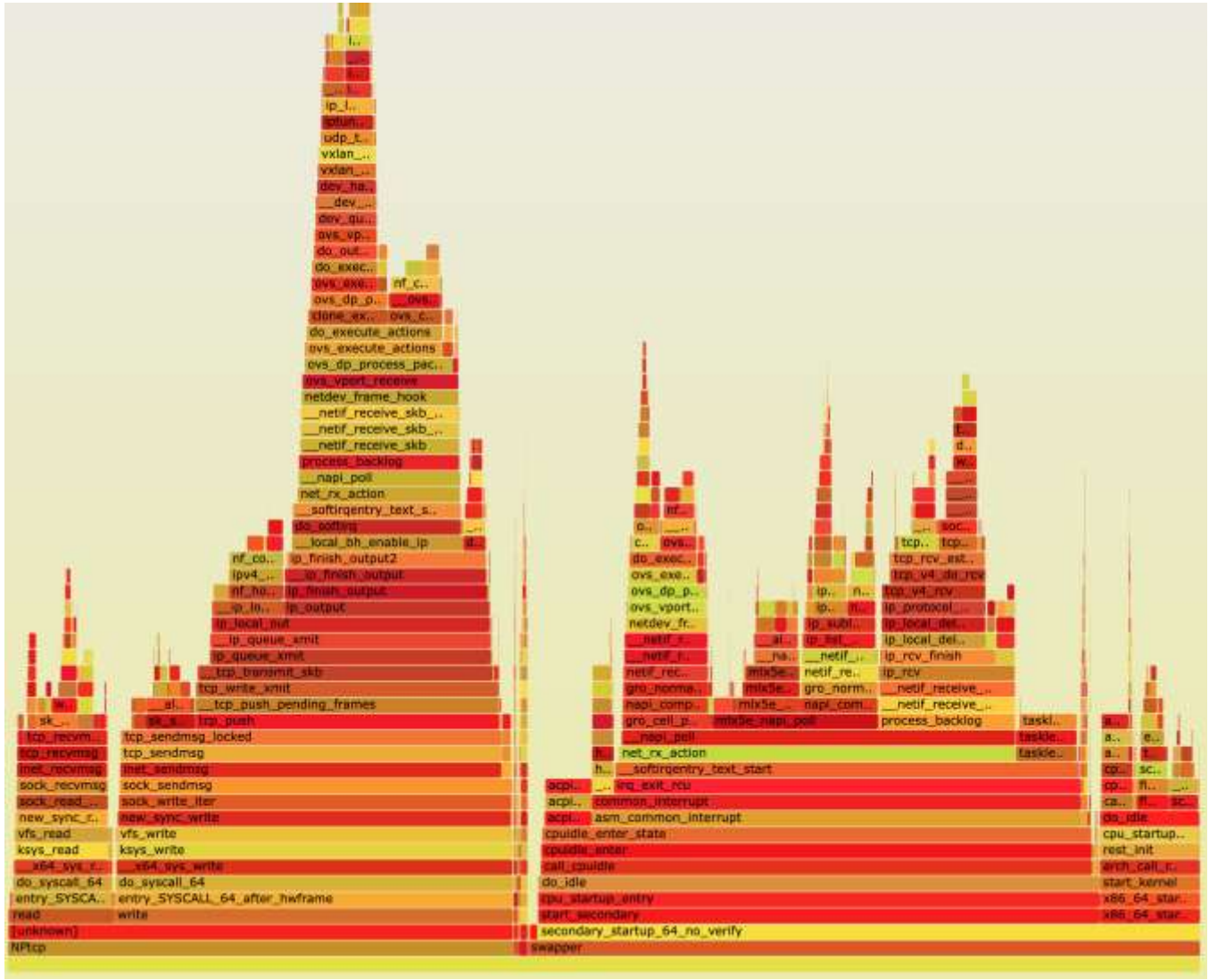
贡献一：定量分解Overlay网络开销

- **定量分解容器网络开销：**

Step 1: 利用火焰图分析overlay网络开销来源

Step 2: 基于eBPF抓取关键函数执行时长





贡献一：定量分解Overlay网络开销

➢ 定量分解容器网络开销：

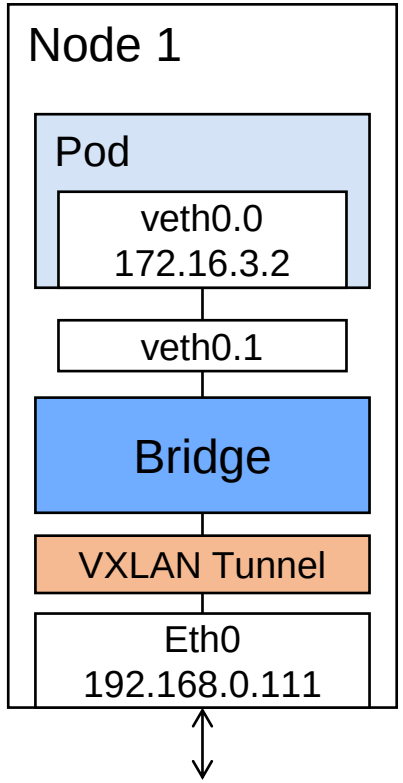
Step 1: 利用火焰图分析overlay网络开销来源

Step 2: 基于eBPF抓取关键函数执行时长

➢ 定量分解结果：

✓ Overlay网络相关开销占据47%的egress路径开销；44%的ingress路径开销。

✓ 容器Overlay网络确实会带来显著CPU开销，从而影响性能。



	Egress					Ingress				
Data path	Overhead type	Antrea	Cilium	BM	**Ours**	Overhead type	Antrea	Cilium	BM	**Ours**
Application network stack	skb allocation	1505	1566	1461	**1509**	skb releasing	715	818	780	**714**
	Conntrack	778	0	788	**763**	Conntrack	616	0	600	**592**
	Netfilter	0	0	305	**0**	Netfilter	0	0	173	**0**
	Others	423	560	547	**519**	Others	838	1016	979	**982**
Veth pair*	NS traversing	562	594		**489**	NS traversing	400			
eBPF*	eBPF		1513		**511**	eBPF		1429		**289**
Open vSwitch*	Conntrack	872				Conntrack	758			
	Flow matching	354				Flow matching	308			
	Action execution	92				Action execution	66			
VXLAN network stack*	Conntrack	0	471			Conntrack	0	271		
	Netfilter	667	421			Netfilter	466	303		
	Routing	50	468			Routing	294	554		
	Others	319	127			Others	619	444		
Link layer	Link layer	1858	1763	1799	**1700**	Link layer	2790	2848	2800	**2737**
Sum		7479	7483	4900	**5491**		7869	7683	5332	**5315**
Latency (μs)		22.97	23.15	16.57	**17.49**		22.97	23.15	16.57	**17.49**

贡献二：在容器网络引入Cache设计

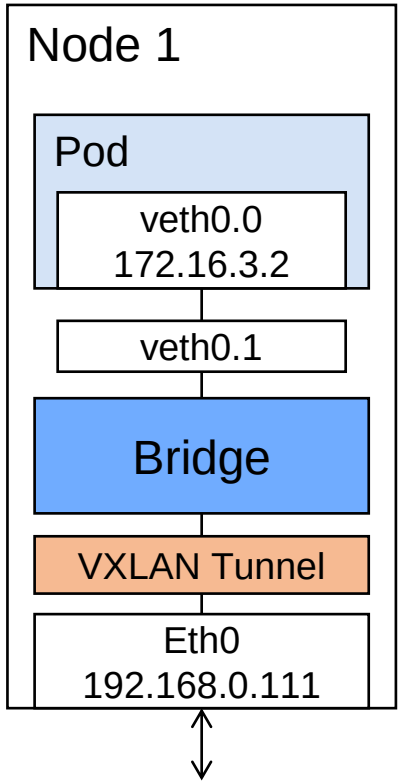
➢ 挑战

用极低开销实现完整overlay网络功能

➢ 贡献

✓ 总结出overlay开销的“不变性”

✓ 在容器网络引入Cache设计



VXLAN路由与报文封装，机内转发

- 处理主要基于IP地址

➡ **发向同一目的容器IP的报文处理结果相同**

报文过滤与防火墙

- 处理主要基于“流”
  由源目端口+传输协议+源目IP定义

➡ **属于同一条“流”的报文处理结果相同**

贡献二：在容器网络引入Cache设计

➢ 挑战

用极低开销实现完整overlay网络功能

➢ 贡献

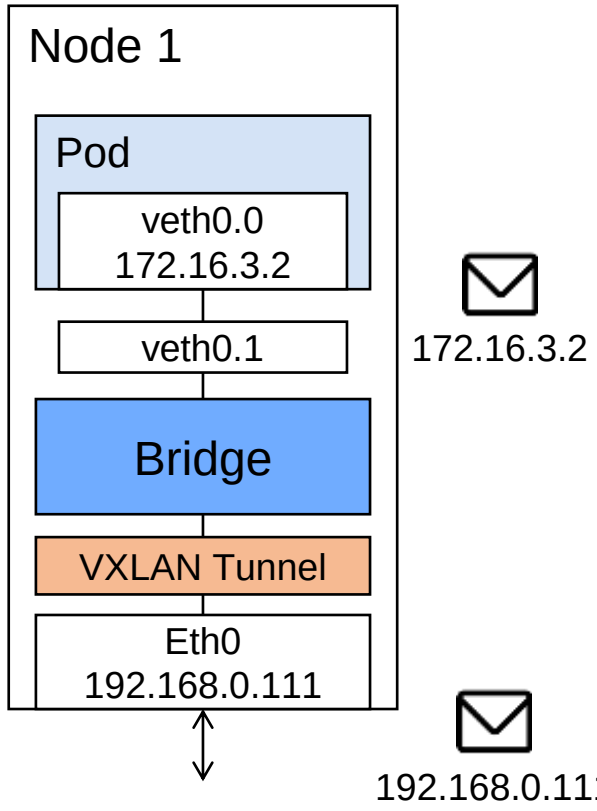
✓ 总结出overlay开销的“不变性”

✓ 在容器网络引入Cache设计

每条流首个报文

“Cache miss”

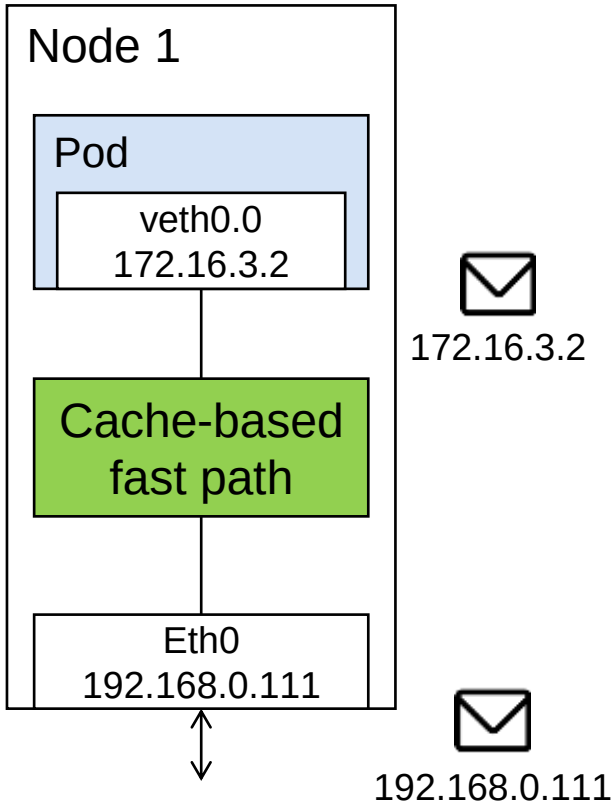
使用原始overlay网络处理，同时存下cache



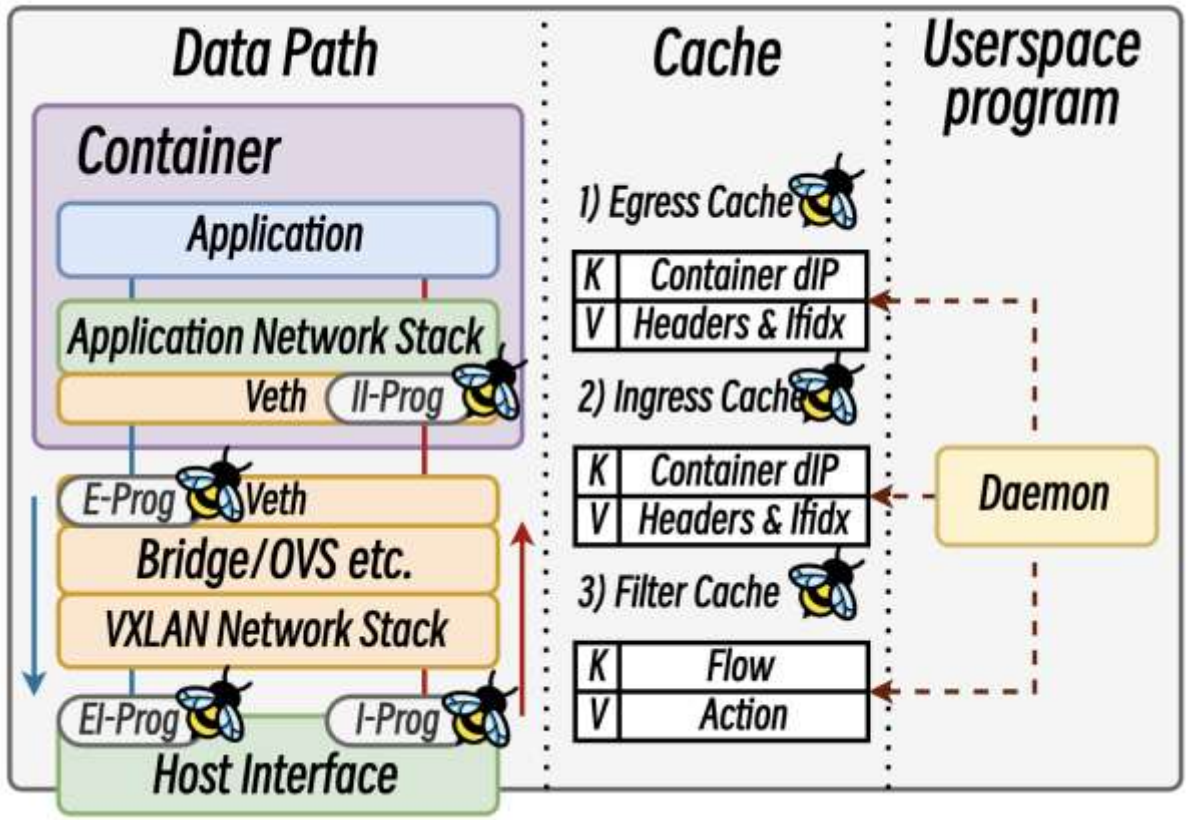
后续报文

“Cache hit”

用查一次cache代替原本overlay网络中的开销



贡献三：基于eBPF的轻量化实现



- eBPF技术是一种允许用户在内核态安全运行程序Linux内核技术

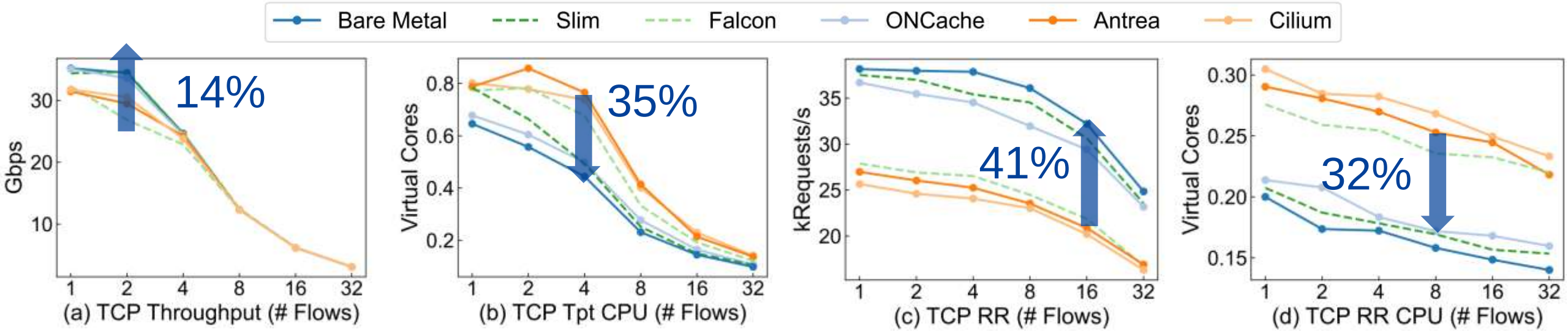
- **挑战**
  - 容器overlay网络在Linux kernel中实现
  - 修改kernel工程量大且难以实际部署

- **贡献**
  - ✓ 提出用eBPF技术轻量化地实现基于cache的overlay网络
  - ✓ 用eBPF map在内存中实现cache
  - ✓ 将eBPF程序挂载在内核处理报文的路径上，利用cache快速处理报文
  - ✓ 避免修改kernel，易于在生产环境部署

实验结果：Microbenchmark

TCP吞吐

TCP RR
(TCP Request-Response)



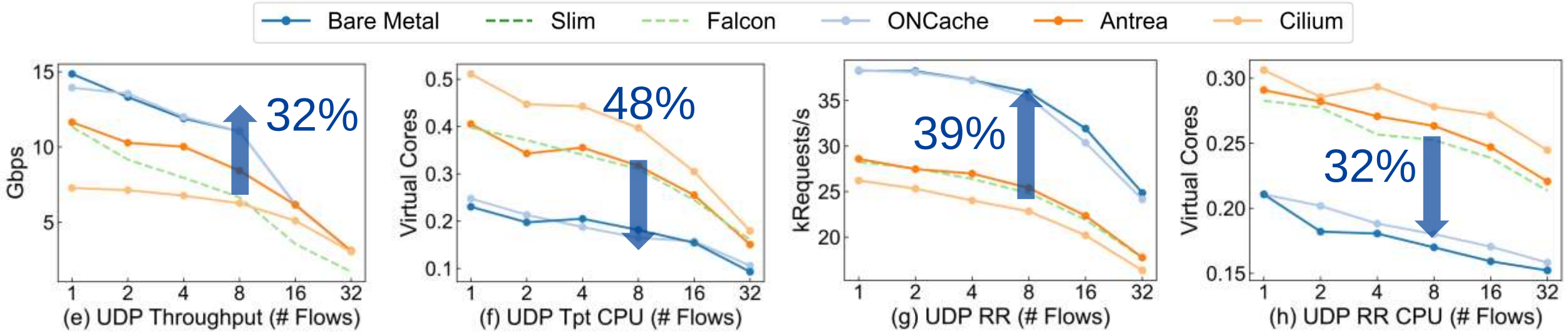
- 吞吐：提高12%-14%
- Per-byte CPU占用：降低14%-35%

- RR：提高36%-41%
- Per-packet CPU占用：降低26%-32%

实验结果：Microbenchmark

UDP 吞吐

UDP RR
(UDP Request-Response)



- 吞吐：提高20%-32%
- Per-byte CPU占用：降低30%-48%

- RR：提高34%-39%
- Per-packet CPU占用：降低28%-32%

ONCache落地需要的额外处理

网络TCP连接的释放优化

开启 ONCache 功能后，Pod 间的网络流量 bypass 的主机的 conntrack，节点上 TCP 连接依旧显示为 established，不会被系统清理，额外占用了系统资源。

节点间 VXLAN 访问优化

开启 ONCache 功能的节点和未开启 ONCache 功能的节点间通信会受到影响，原因是双向通信报文不在同一条路径（发送报文基于标准vxlan，回复报文基于“快速路径”）。

节点和 Pod 间访问优化

主机和容器网络的 Pod 间通信会受到影响，原因是目前 ONCache 针对 Pod 的 veth 设备进行处理，但节点的虚拟设备尚未兼容，导致双向通信报文不在同一条路径。

解决方案

识别到FIN/RST报文，直接走标准 VXLAN 路径进行传输，确保节点上TCP连接能正确关闭。

提前缓存开启 ONCache 的节点和对应 Pod 网段，只有在缓存内的网段才使用“快速路径”进行传输。

当 SourceIP 为网段起始地址（例如 10.224.0.1）时，直接走标准 VXLAN 路径进行传输。

数据库对 ONCache 的需求

TRANSWARP 星环科技 | 股票代码 688031

星环知识管理平台是星环科技推出的人工智能基础设施产品，为企业提供从语料到模型再到应用的全链路 AI 工具集，其底层涉及到了：【向量数据库】，【分析性数据库】，【图数据库】，【时序数据库】。

数据库与客户端通信优化

数据库与客户端之间的通信需要高效的网络支持，ONCache 技术可以实现对客户端请求的快速响应和数据的高效传输。

分布式事务的通信协作

在分布式数据库中，事务处理需要多个节点之间的通信协作，ONCache技术可以优化这些通信过程，提高分布式事务性能。

数据库性能提升的具体表现

1

查询性能提升

应用 ONCache 技术后，数据库查询性能显著提升，查询性能提升了 30% 左右。这是由于ONCache 优化了网络通信路径，减少了数据传输延迟。

例如，在处理大规模并发数据查询时，ONCache 技术能够支撑更高的并发查询，提高了查询效率。

2

数据传输效率提升

ONCache 技术通过优化网络协议栈，减少了数据在传输过程中的损耗和延迟，使得数据库的数据传输效率也得到了进一步的提升。

在多主复制架构中，每个写请求都会在本地处理后同步到其他节点，网络性能的提升使得同步更加高效，用户感受到的延迟更小

3

系统资源利用率提升

由于 ONCache 技术减少了 CPU 对 VXLAN 报文的封装，使得数据库系统对 CPU 资源利用率得到了提高。

这使得数据库能够在有限的硬件资源下处理更多的并发请求，提高了系统的整体性能和可扩展性。

数据库性能提升的具体表现

测试命令：

sysbench --threads=50 --report-interval=3 --time=30 --db-ps-mode=auto --percentile=99 --auto_inc=off --tables=100 --table_size=100000 oltp_point_select run

- 标准VXLAN-QPS： ***8.5W***

```
    total:                          2552925
transactions:                       2552925  (85055.34 per sec.)
queries:                            2552925  (85055.34 per sec.)
ignored errors:                     0        (0.00 per sec.)
reconnects:                         0        (0.00 per sec.)
```

- ONCache-QPS： ***10.78W***

```
    total:                          3237602
transactions:                       3237602  (107867.86 per sec.)
queries:                            3237602  (107867.86 per sec.)
ignored errors:                     0        (0.00 per sec.)
reconnects:                         0        (0.00 per sec.)
```

- 主机网络-QPS： ***10.07W***

```
    total:                          3023335
transactions:                       3023335  (100706.60 per sec.)
queries:                            3023335  (100706.60 per sec.)
ignored errors:                     0        (0.00 per sec.)
reconnects:                         0        (0.00 per sec.)
```

提升26.8%

长期愿景

NodePort的支持

未来，ONCache 技术可以扩展到 NodePort 场景，实现对 NodePort 的高效处理和优化，提高 NodePort 的性能和可靠性。

在容器编排环境中，NodePort 是常见的网络通信方式，ONCache 技术的应用将为 NodePort 通信提供更强大的支持。

多VPC环境下的支持

ONCache 技术目前仅支持 NodeIPAM 类型的 CNI，未来可以支持基于 GlobalIPAM 的多 VPC 环境，以适应不同的环境。

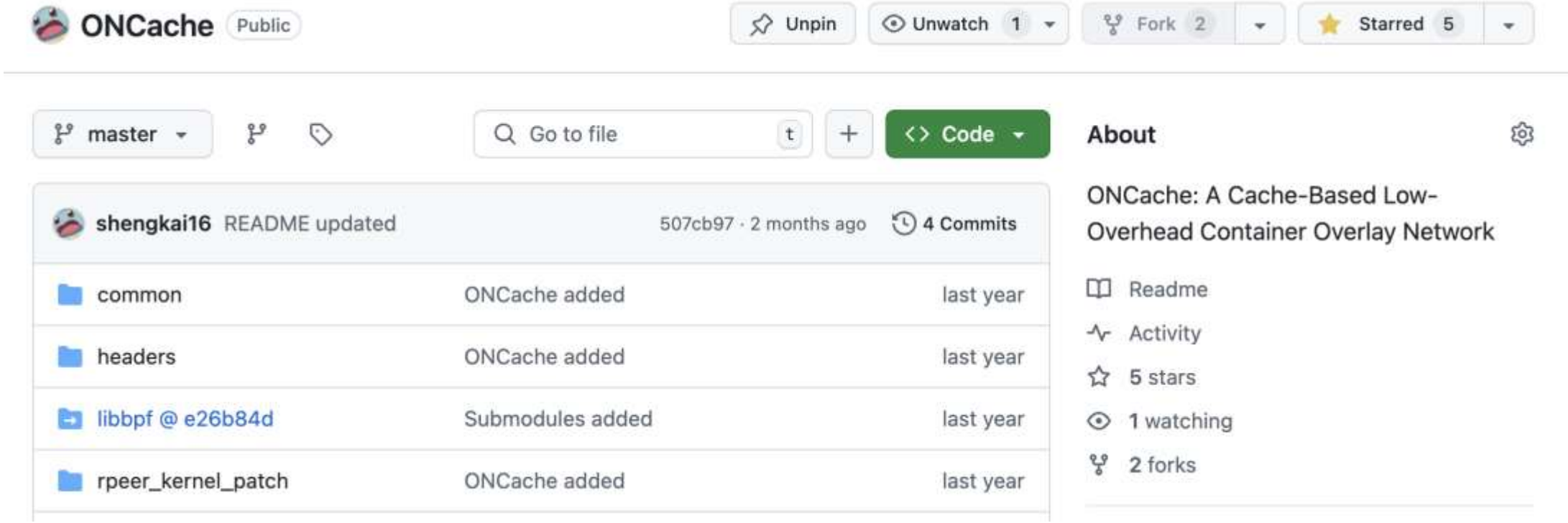
在云计算环境中，多 VPC 环境下的通信优化将为企业提供更灵活的网络部署方案，提高云资源的利用率。

跨集群网络的支持

ONCache 技术目前仅支持 k8s 集群内的 VXLAN 网络加速，未来可以支持多个集群间的 VXLAN 网络加速。

随着 k8s 的广泛应用，客户部署多个 k8s 集群是一个比较常见的事情，应用跨集群部署也是一种常态。

长期愿景



https://github.com/shengkai16/ONCache

- **长期愿景**
  - ✓ 推动ONCache在生产环境中广泛落地
  - ✓ 成为容器集群网络的新标准