KubeCon | CloudNativeCon

China 2018

Benchmarking Various CNI Plugins

Giri Kuncoro & Vijay Dhama from GOJEK

Agenda

- Overview of Various CNI Plugins
- Experiments
    - Goals
    - Environment
    - Results
- Takeaways

KubeCon | CloudNativeCon

China 2018

Overview of Various CNI Plugins

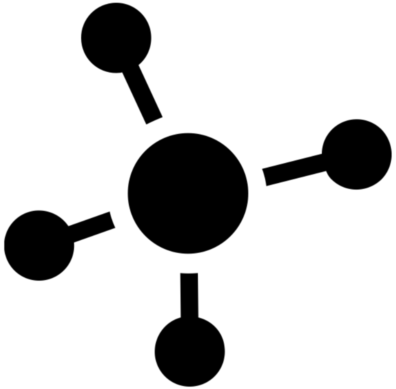
Kubernetes Network Model

- All containers communicate **without NAT**
- All nodes communicate with containers **without NAT**
- Container sees its own IP as others see it

Kubernetes **doesn't provide default network implementation**, it leaves it to **third party tools**

source: https://kubernetes.io/docs/concepts/cluster-administration/networking/#kubernetes-model

What CNI do?



Connectivity

Reachability

CNI Plugins

Project Calico - a layer 3 virtual network
Weave - a multi-host Docker network
Contiv Networking - policy networking for various use cases
SR-IOV
Cilium - BPF & XDP for containers
Infoblox - enterprise IP address management for containers
Multus - a Multi plugin
Romana - Layer 3 CNI plugin supporting network policy for Kubernetes
CNI-Genie - generic CNI network plugin
Nuage CNI - Nuage Networks SDN plugin for network policy kubernetes support
Silk - a CNI plugin designed for Cloud Foundry
Linen - a CNI plugin designed for overlay networks with Open vSwitch and fit in SDN/OpenFlow network environment
Vhostuser - a Dataplane network plugin - Supports OVS-DPDK & VPP
Amazon ECS CNI Plugins - a collection of CNI Plugins to configure containers with Amazon EC2 elastic network interfaces (ENIs)
Bonding CNI - a Link aggregating plugin to address failover and high availability network
ovn-kubernetes - an container network plugin built on Open vSwitch (OVS) and Open Virtual Networking (OVN) with support for both Linux and Windows
Juniper Contrail / TungstenFabric - Provides overlay SDN solution, delivering multicloud networking, hybrid cloud networking, simultaneous overlay-underlay support, network policy enforcement, network isolation, service chaining and flexible load balancing
Knitter - a CNI plugin supporting multiple networking for Kubernetes

source: https://github.com/containernetworking/cni#3rd-party-plugins

Scope

So many CNI plugins to test, limit scope to:

- Flannel
- Calico
- Weave
- Cilium
- Kube-Router
- AWS CNI
- Kopeio
- Romana

Flannel

Simple way to configure L3 network fabric with VXLAN as default

source: https://github.com/coreos/flannel

Calico

Pure L3 approach which enables unencapsulated networks and BGP peering

source: https://projectcalico.org

Weave

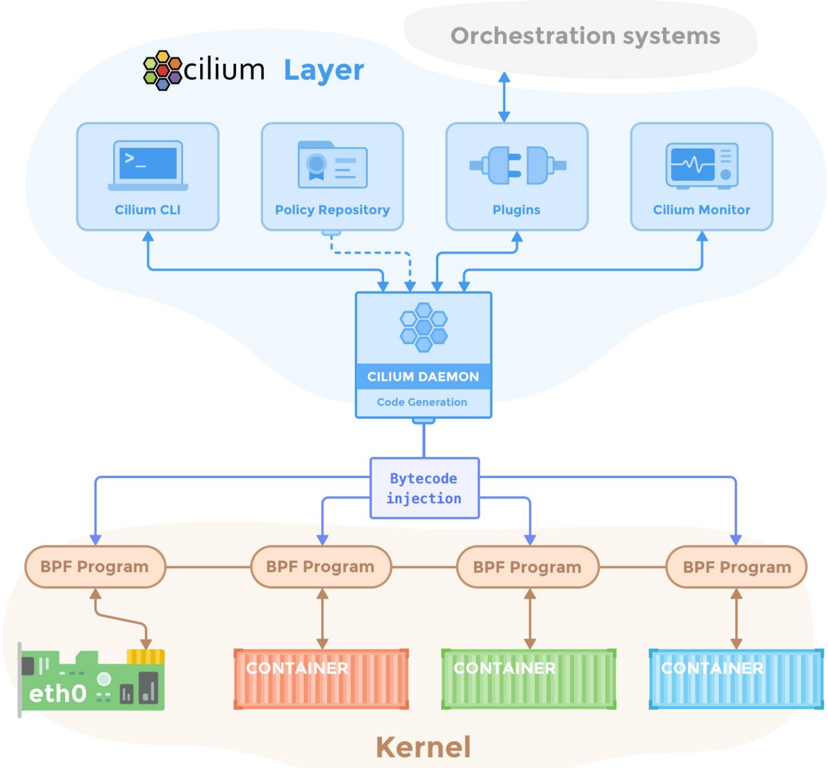
Support overlay network with different cloud network config

weaveworks

source: https://www.weave.works/docs

Cilium

Based on Linux kernel technology called **BPF**



source: https://cilium.readthedocs.io

Kube-Router

Built on standard Linux networking toolset: ipset, iptables, IPVS, LVS

KUBE ROUTER

source: https://github.com/cloudnativelabs/kube-router

AWS CNI

Using AWS ENI interface for pod networking

source: https://github.com/aws/amazon-vpc-cni-k8s

Kopeio

Simple VXLAN, but also support L2 with GRE and IPSEC

source: https://github.com/kopeio/networking

Romana

Use standard L3, distributed routes with BGP or OSPF

source: https://romana.io

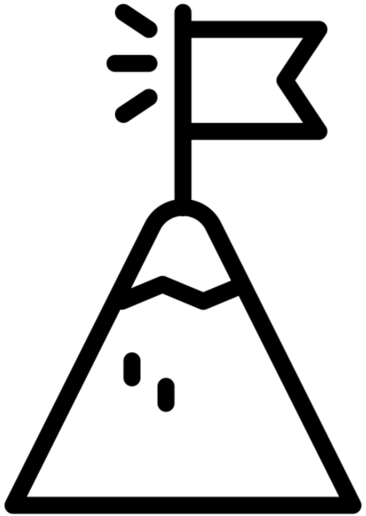
KubeCon | CloudNativeCon

China 2018

Experiments

Goals

- **Lowest latency** and **highest throughput**
- **Different protocols** and various **packet sizes**
- **CPU consumption** and **launch time**
- Kubernetes **network policies**



Environment

- **8 Kubernetes clusters** with different CNIs
- 2 nodes cluster with **m4.xlarge** type by Amazon AWS EC2 with **Debian 9**, **kernel 4.9**
- **Kubernetes** v1.10.9 with **Kops**

Environment

KubeCon CloudNativeCon China 2018

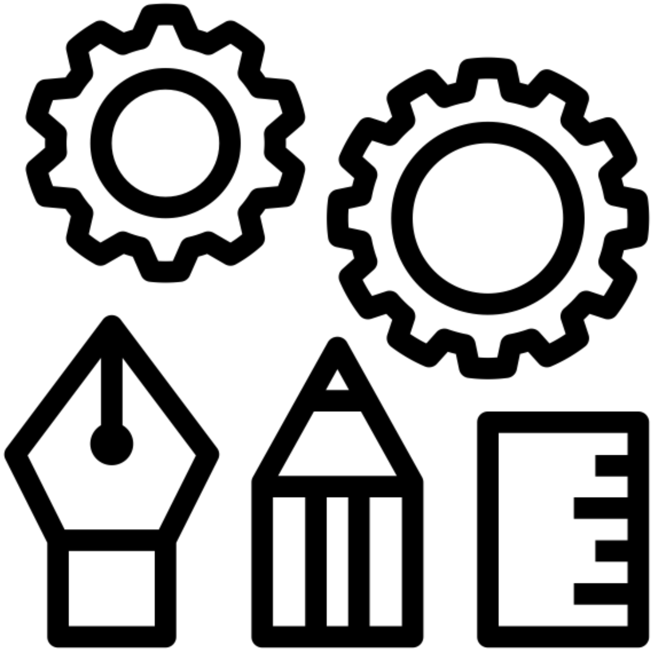
- All CNI plugins deployed with default config in Kops
  - No tuning or custom configuration

  - **Flannel** v0.10.0
  - **Calico** v2.6.7
  - **Weave** v2.4.0
  - **Cilium** v1.0
  - **Kube-Router** v0.1.0
  - **AWS CNI** v1.0.0
  - **Kopeio** v1.0.20180319
  - **Romana** v2.0.2

CNI

GO-JEK

Tools



- Sockperf (v3.5.0)
  - Util over socket API for latency/throughput measurement
- Netperf (v2.6.0)
  - Unidirectional throughput and end-to-end latency measurement
- Tool from PaniNetworks
  - Generate HTTP workloads and measure response

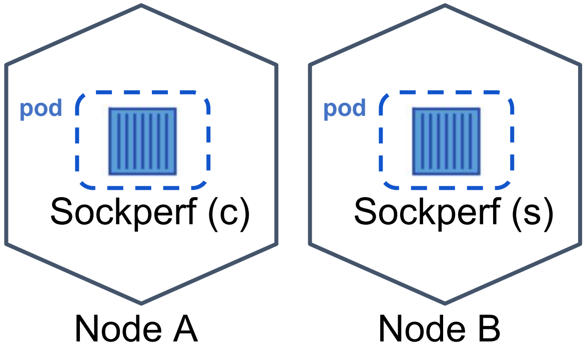
KubeCon | CloudNativeCon

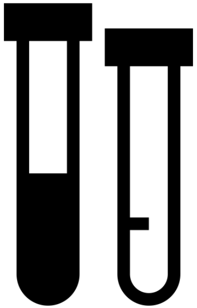
China 2018

Experiment #1

Throughput & Latency

Steps

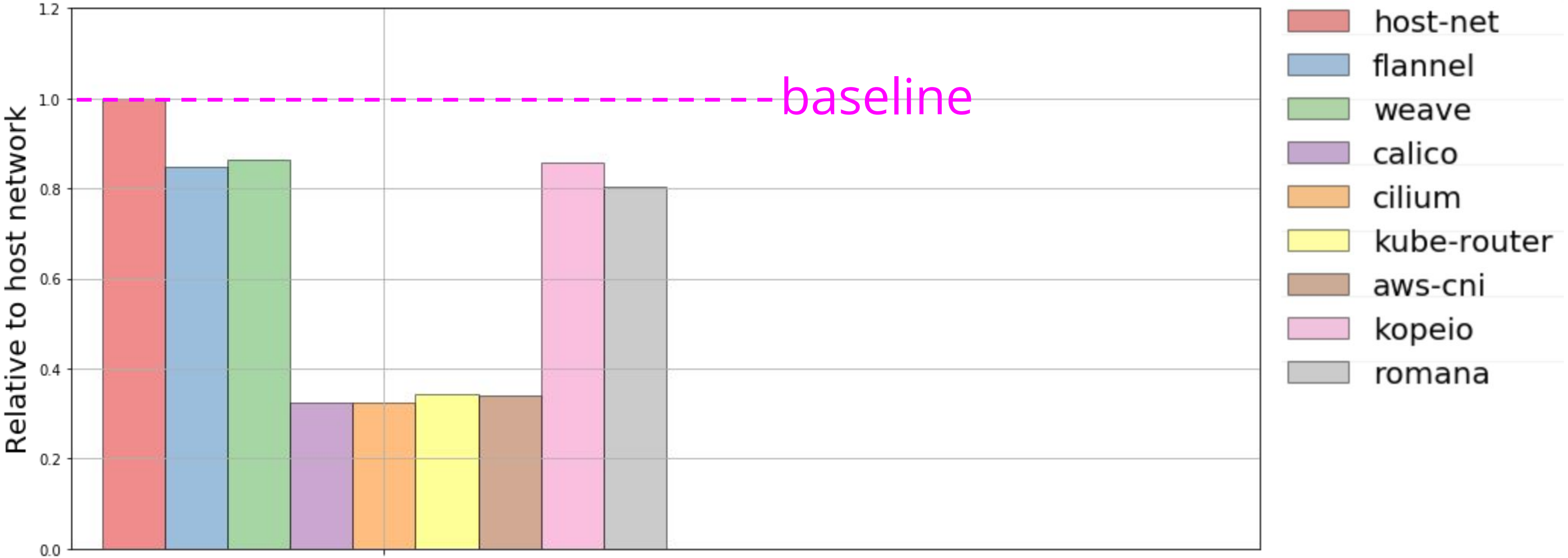




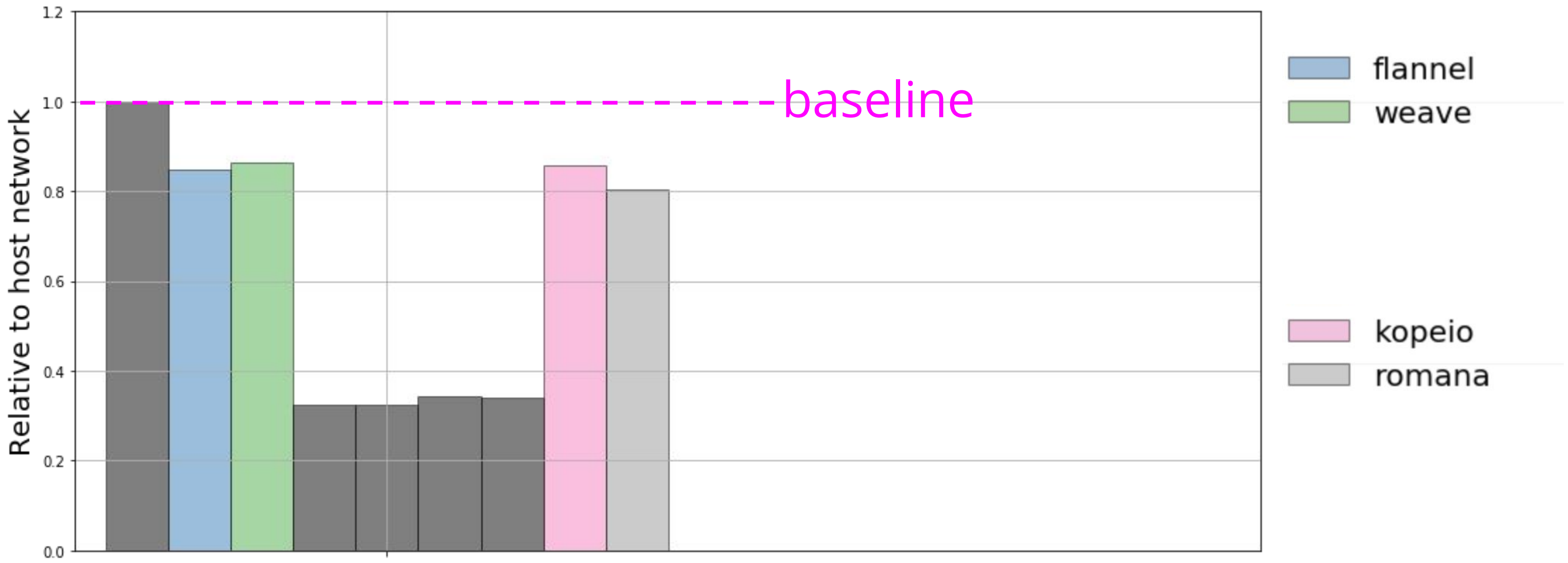
- Sockperf client pod in Node A
- Sockperf server pod in Node B

- 256 bytes for TCP throughput test
- 16 bytes for TCP latency test

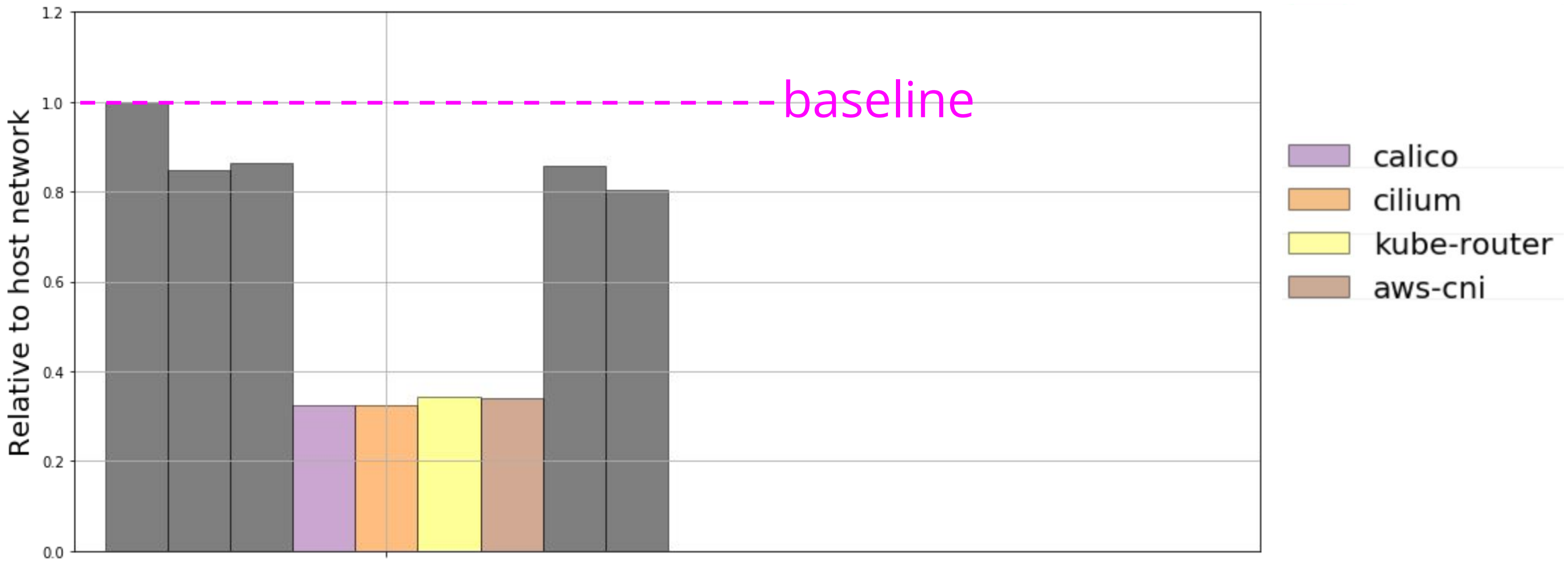
TCP throughput



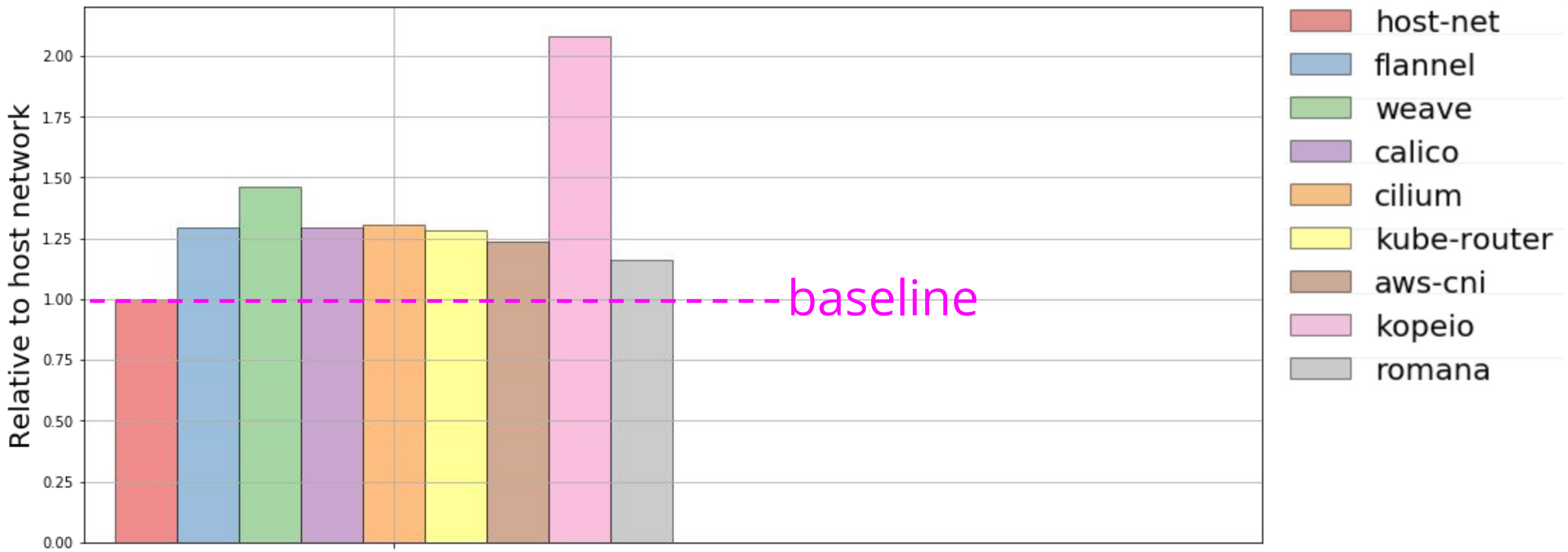
TCP throughput



TCP throughput

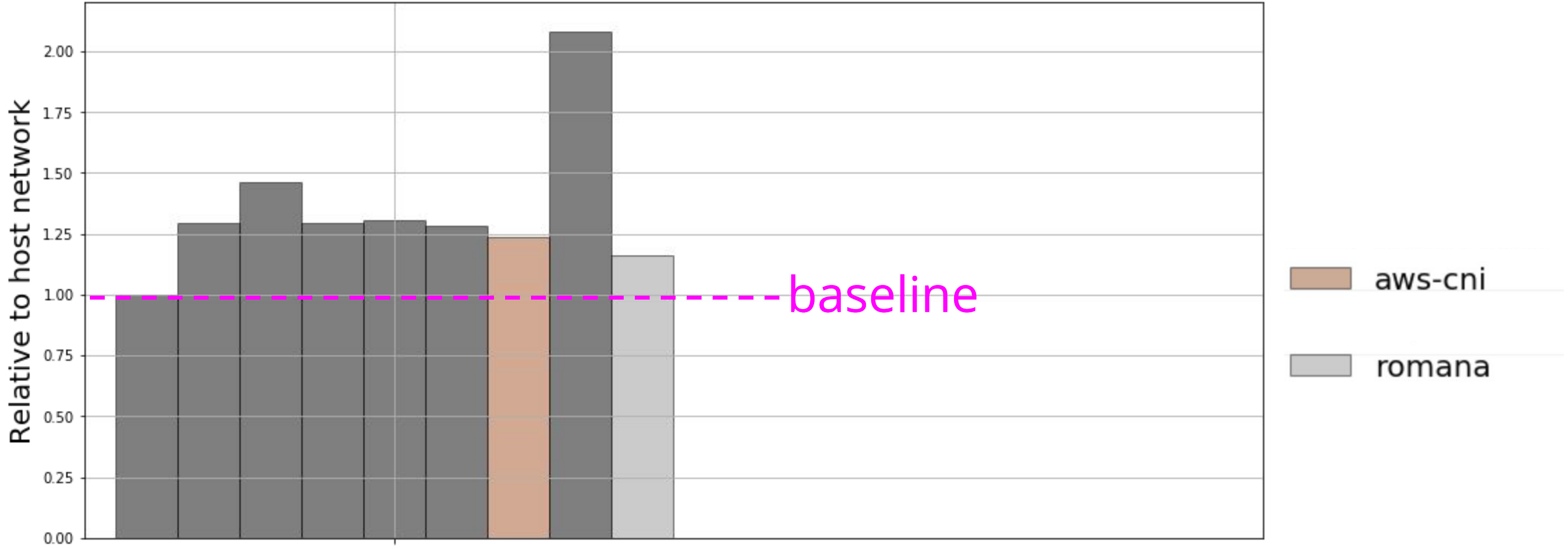


TCP latency



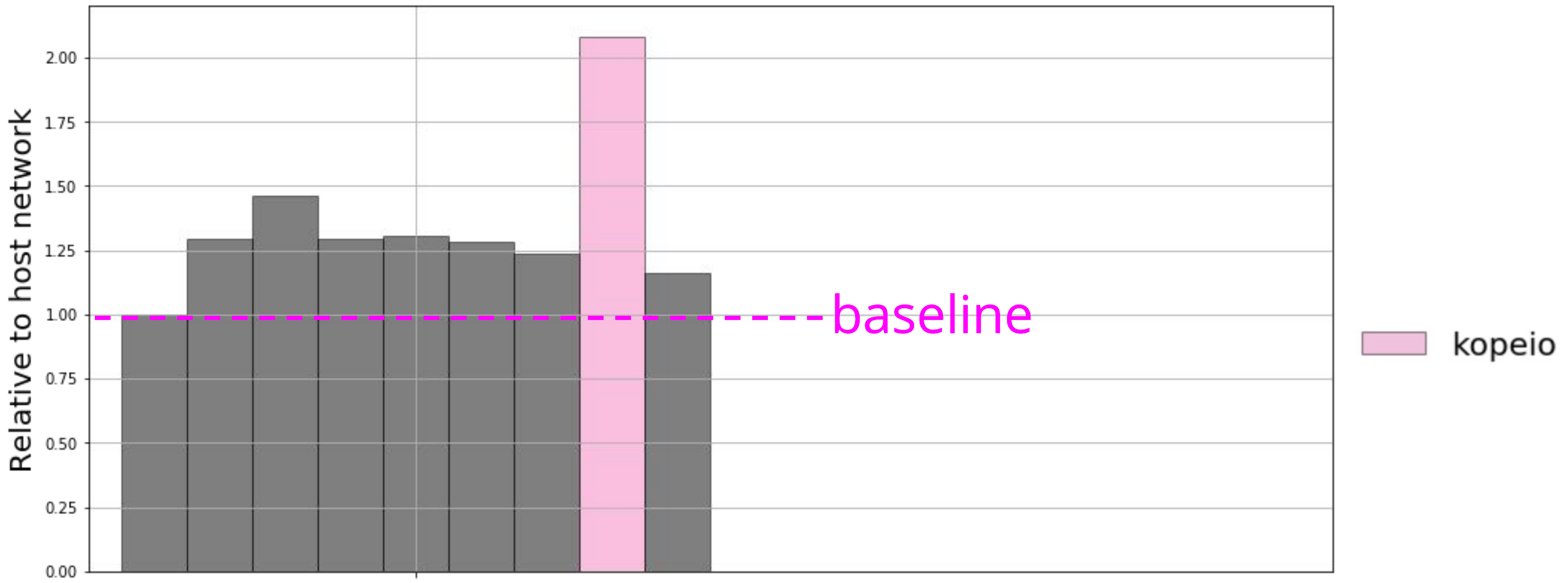
Relative to host network
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
baseline
host-net
flannel
weave
calico
cilium
kube-router
aws-cni
kopeio
romana

TCP latency



Relative to host network
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
baseline
aws-cni
romana

TCP latency



GO JEK

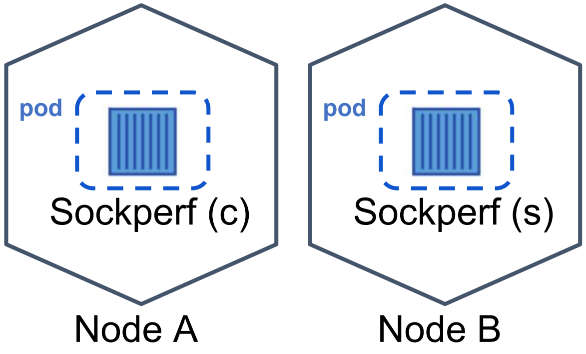
KubeCon | CloudNativeCon

China 2018

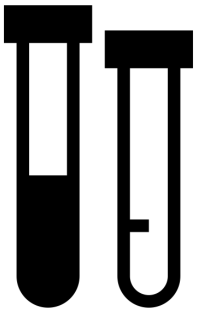
Experiment #2

Protocol & Packet Sizes

Steps

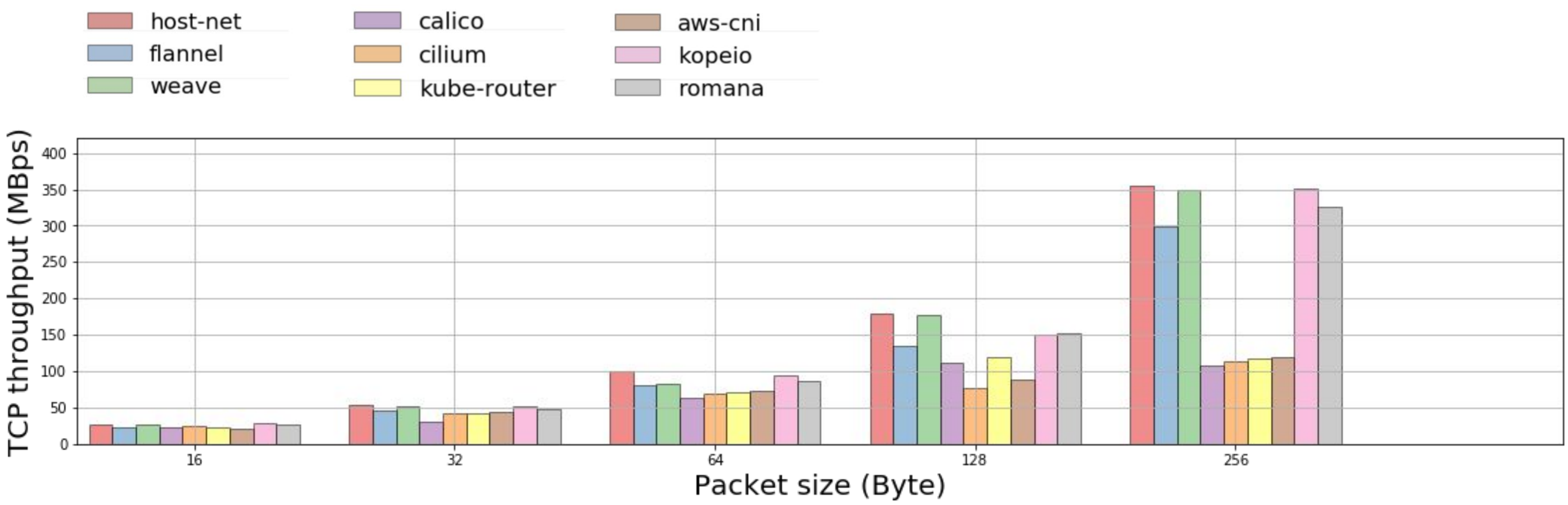


- Sockperf client pod in Node A
- Sockperf server pod in Node B



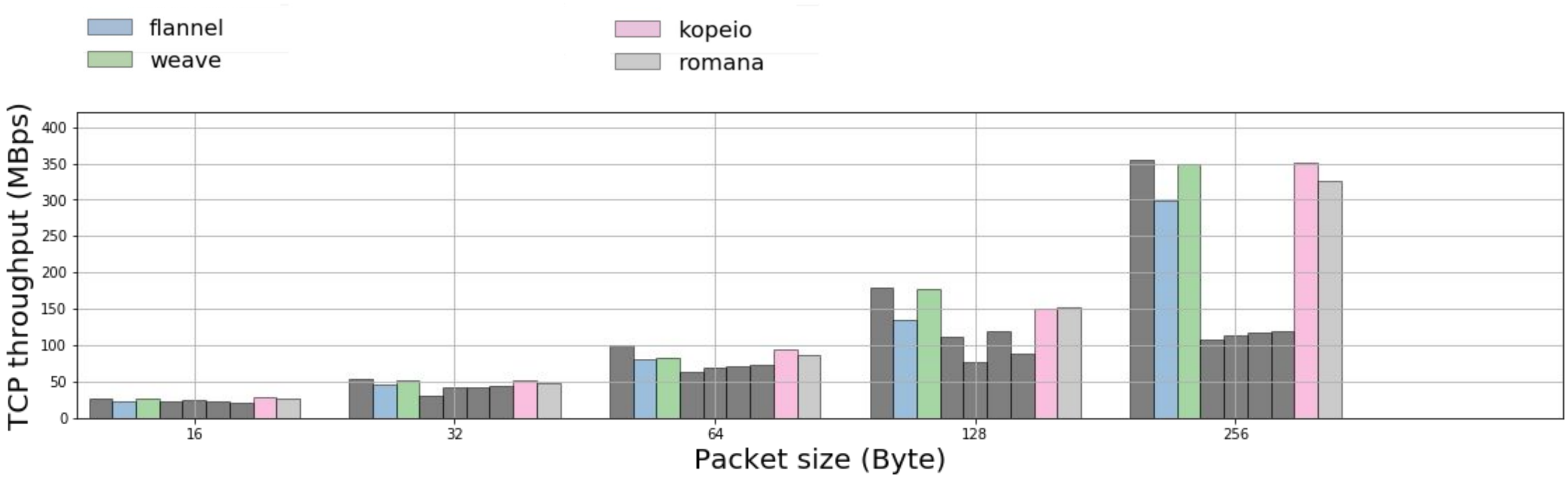
- Measure
  TCP and UDP throughput from 16 to 256 bytes

TCP throughput vs packet sizes

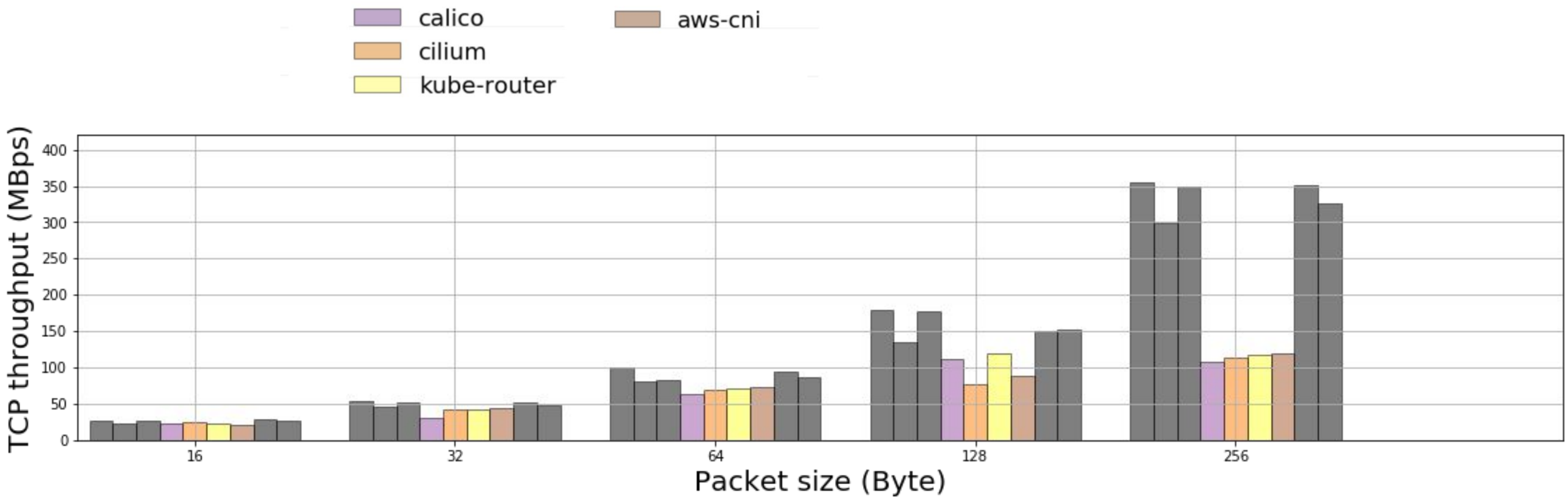


GOJEK

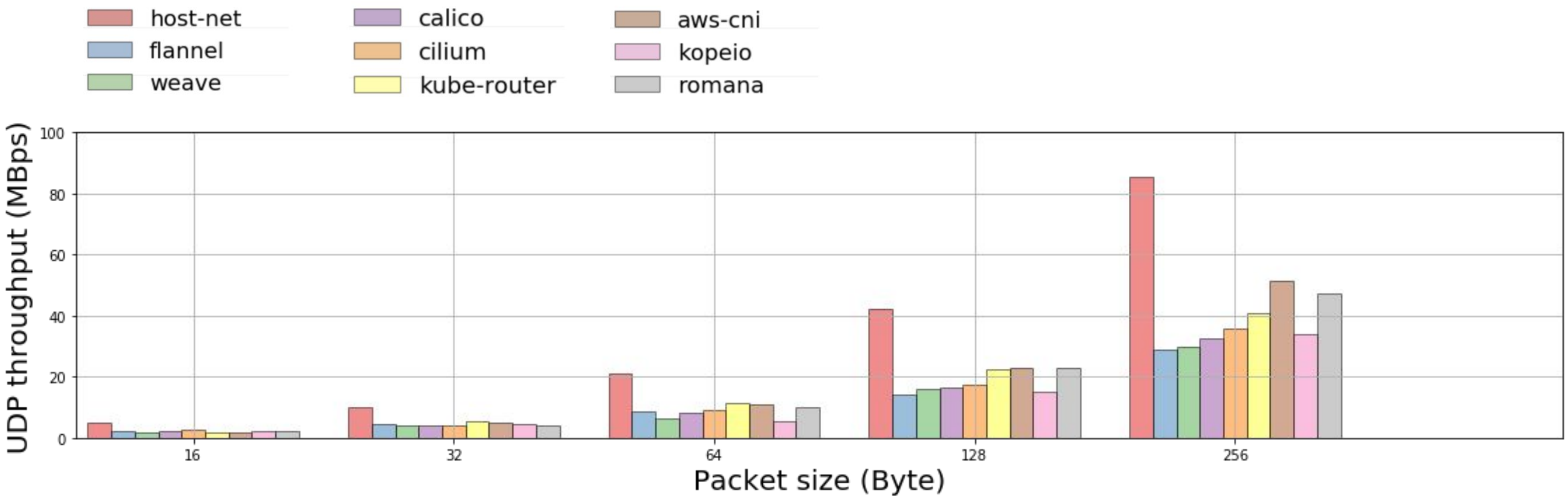
TCP throughput vs packet sizes



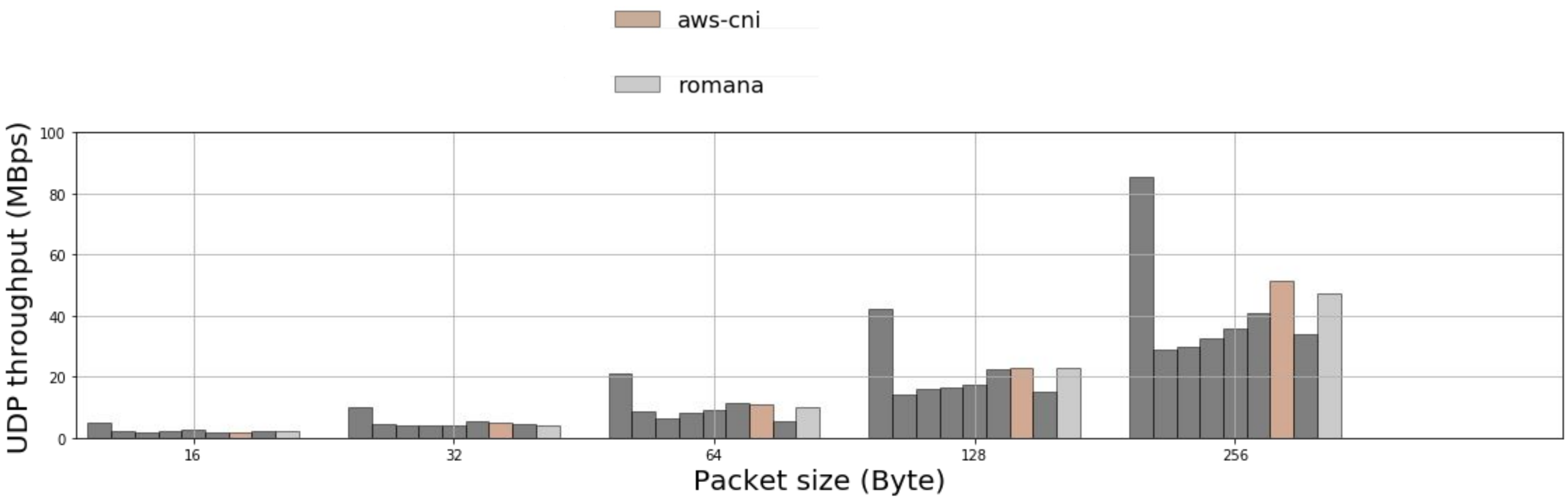
TCP throughput vs packet sizes



UDP throughput vs packet sizes



UDP throughput vs packet sizes



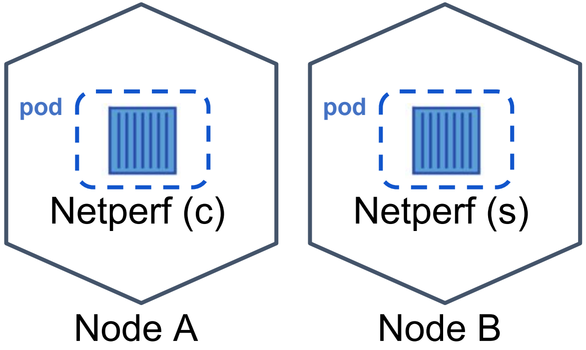
KubeCon | CloudNativeCon

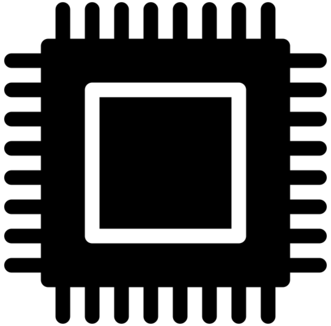
China 2018

Experiment #3

CPU Overhead

Steps

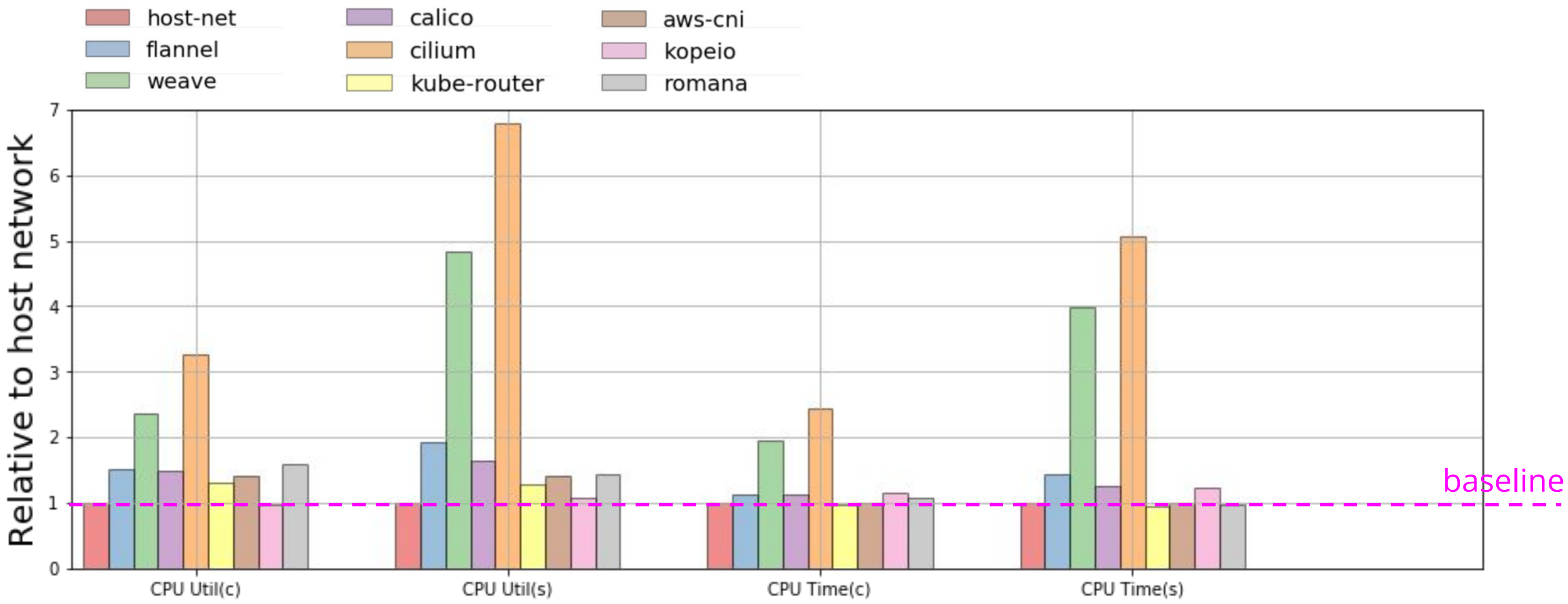




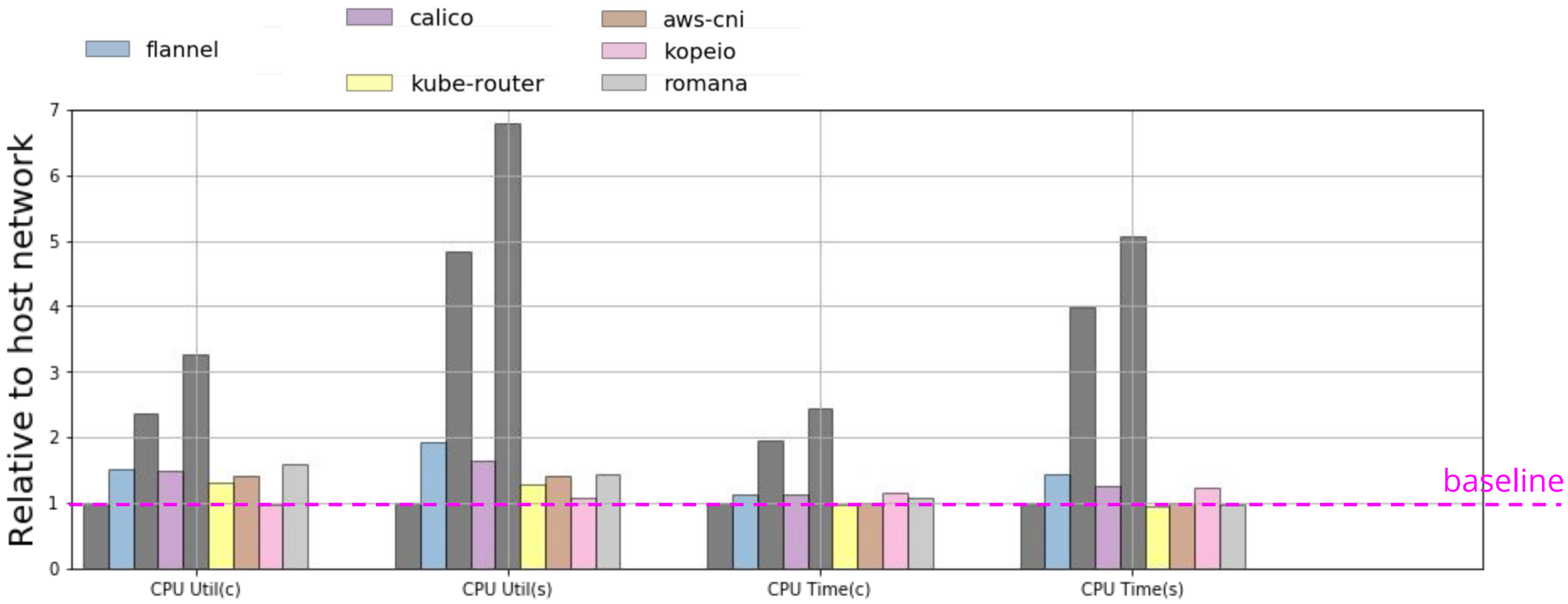
- Netperf client pod in Node A
- Netperf server pod in Node B

- Netperf UDP_RR to measure CPU utilization
- Time spent in user space, kernel space, and waiting for I/O

CPU Overhead



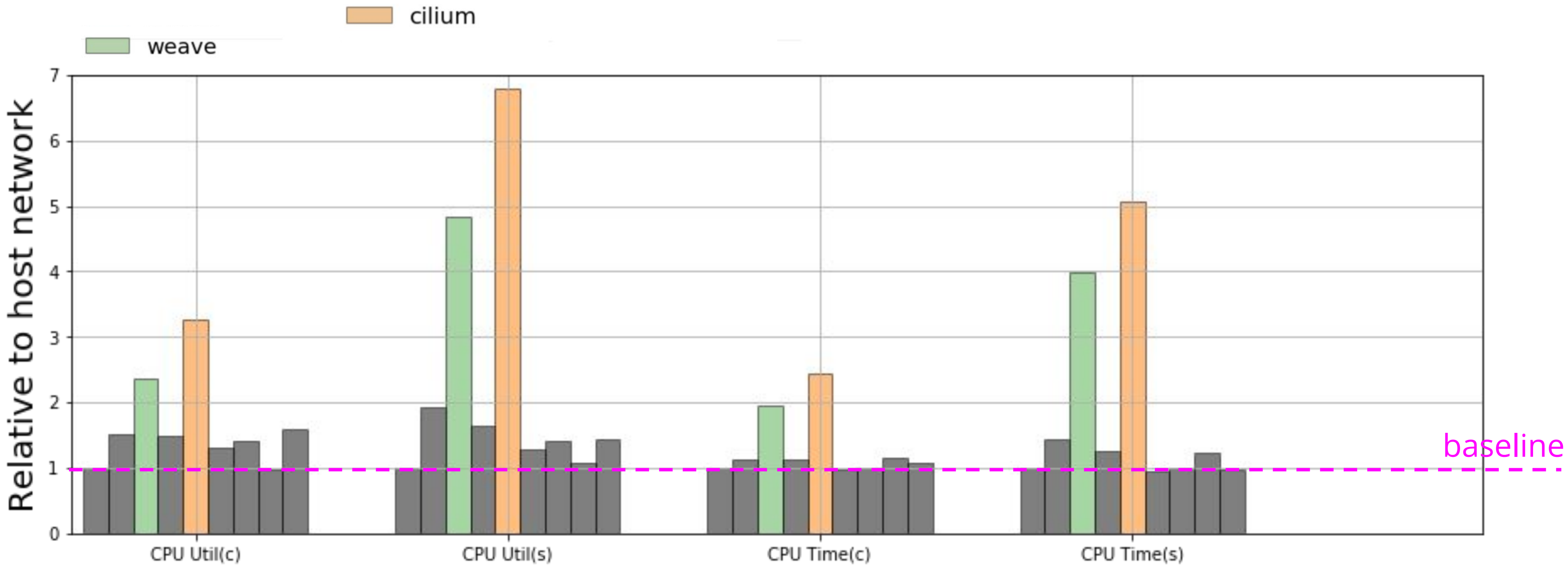
CPU Overhead



KubeCon CloudNativeCon China 2018

GO-JEK

CPU Overhead



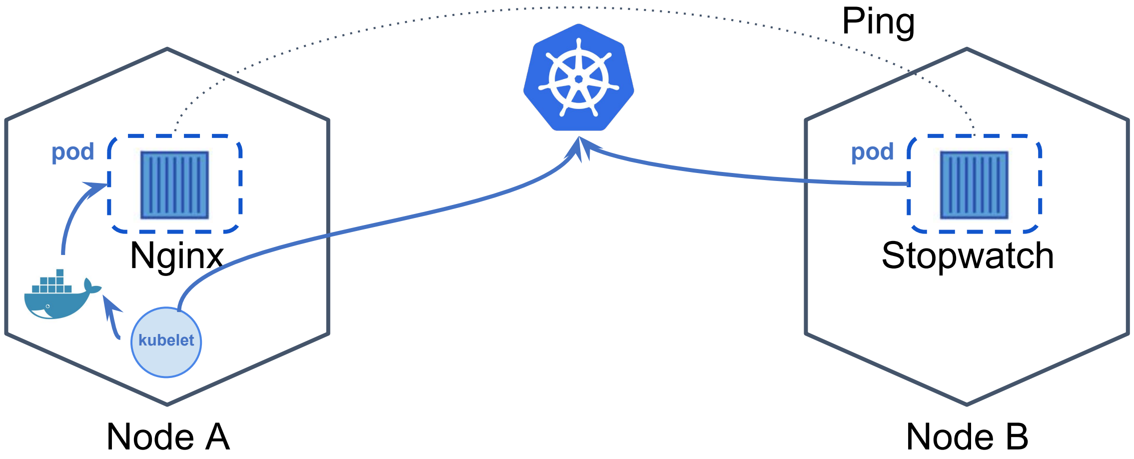
KubeCon | CloudNativeCon

China 2018

Experiment #4

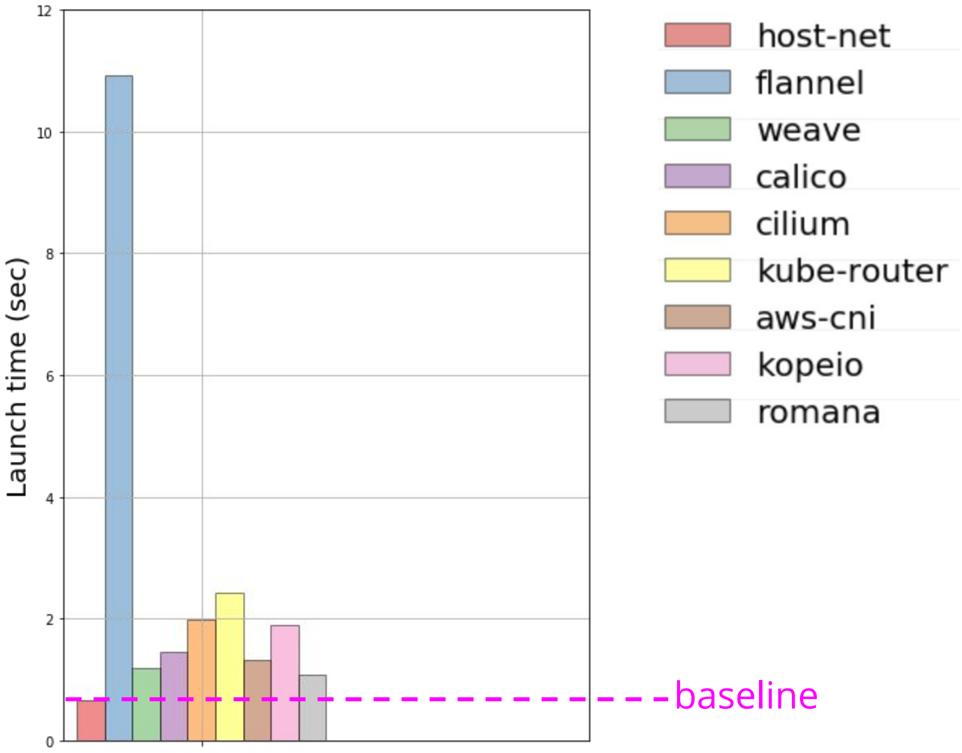
Network Launch Time

Steps

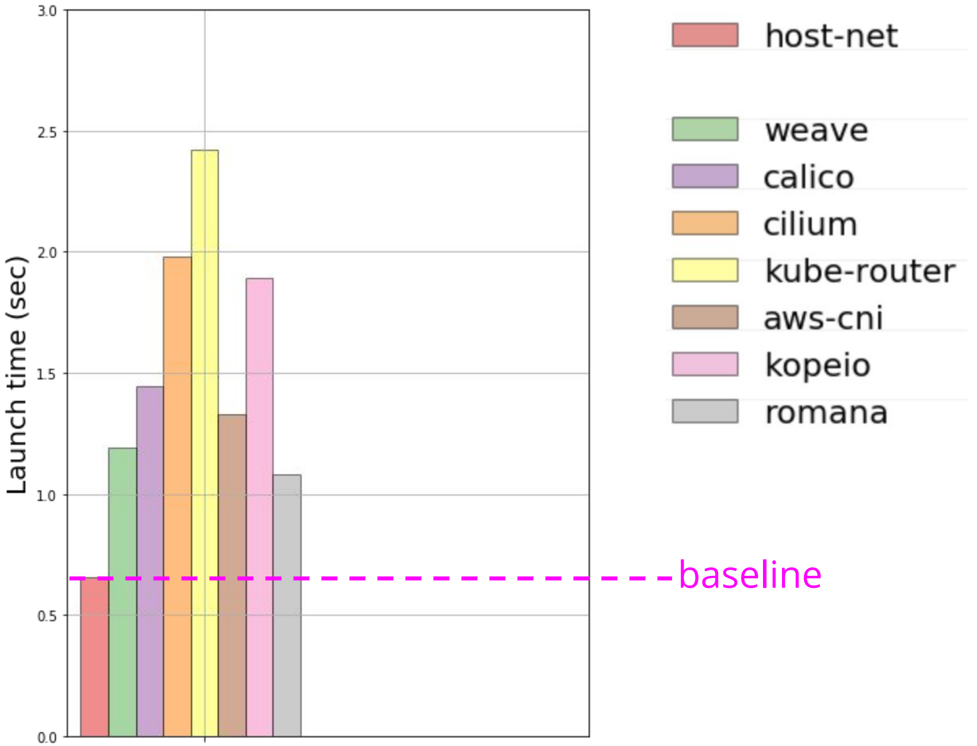


- Deploy Nginx pod in Node A
- Watch kubelet events on container create
- Stopwatch until pinging from Node B

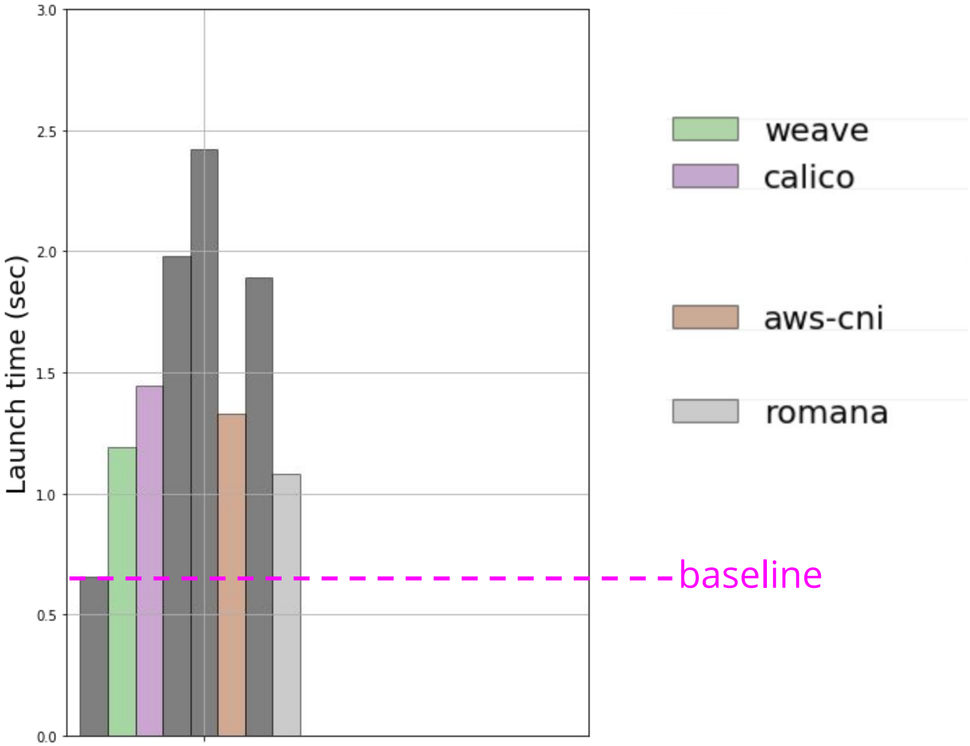
Network launch time



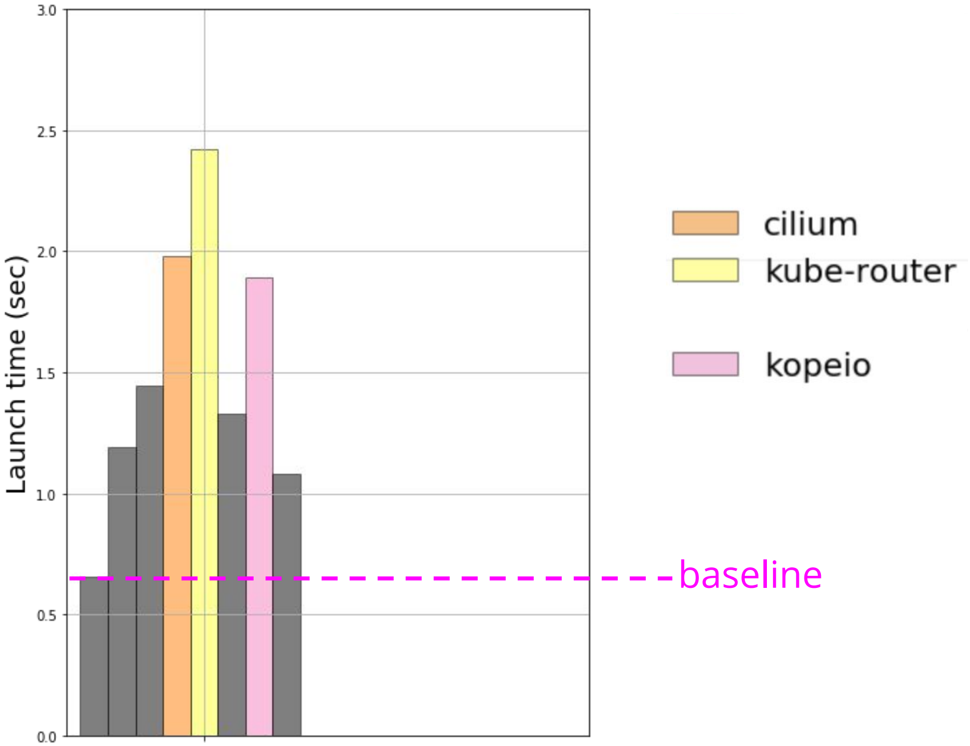
Network launch time



Network launch time



Network launch time



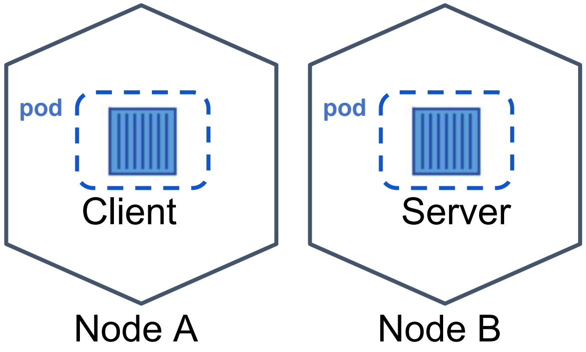
KubeCon | CloudNativeCon

China 2018

Experiment #5

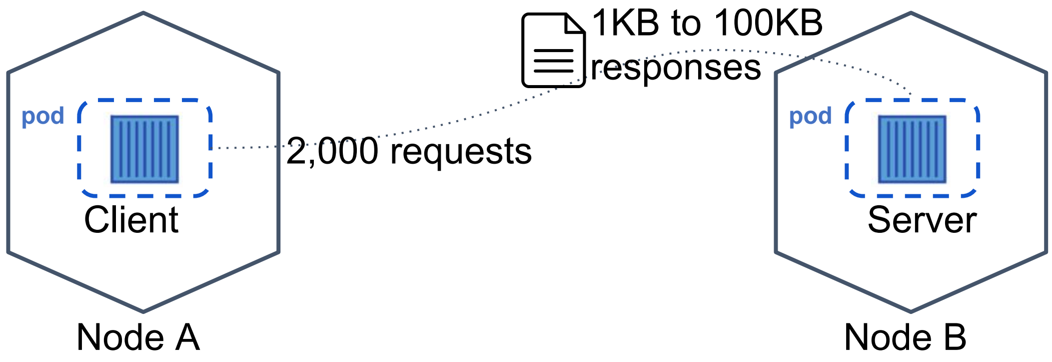
Network Policies

Steps



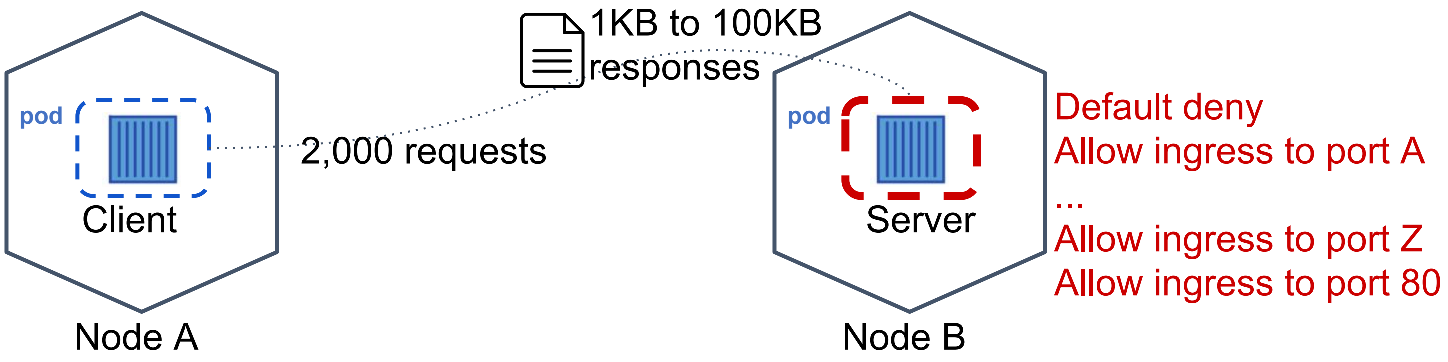
- Client pod in Node A
- Server pod in Node B

Steps



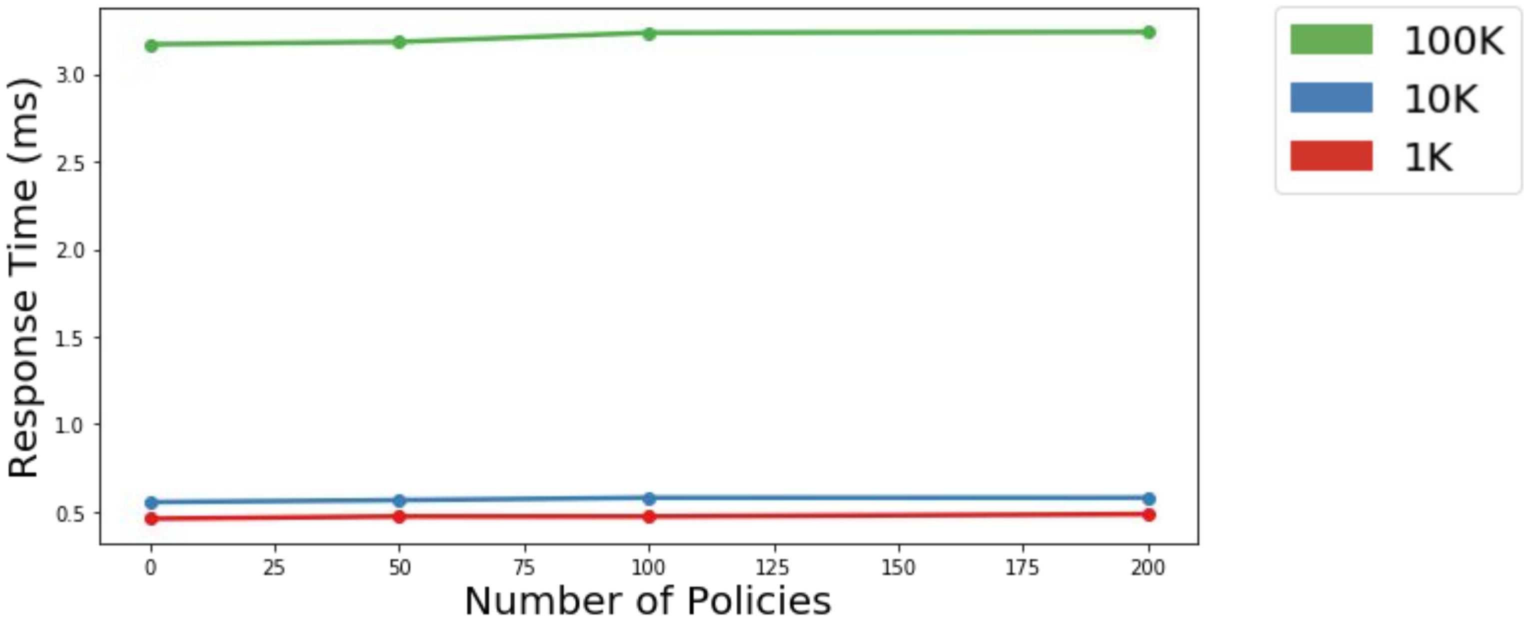
- Client pod sends 2,000 HTTP requests
- Varied response size from 1KB to 100KB
- Disabled persistent connection

Steps

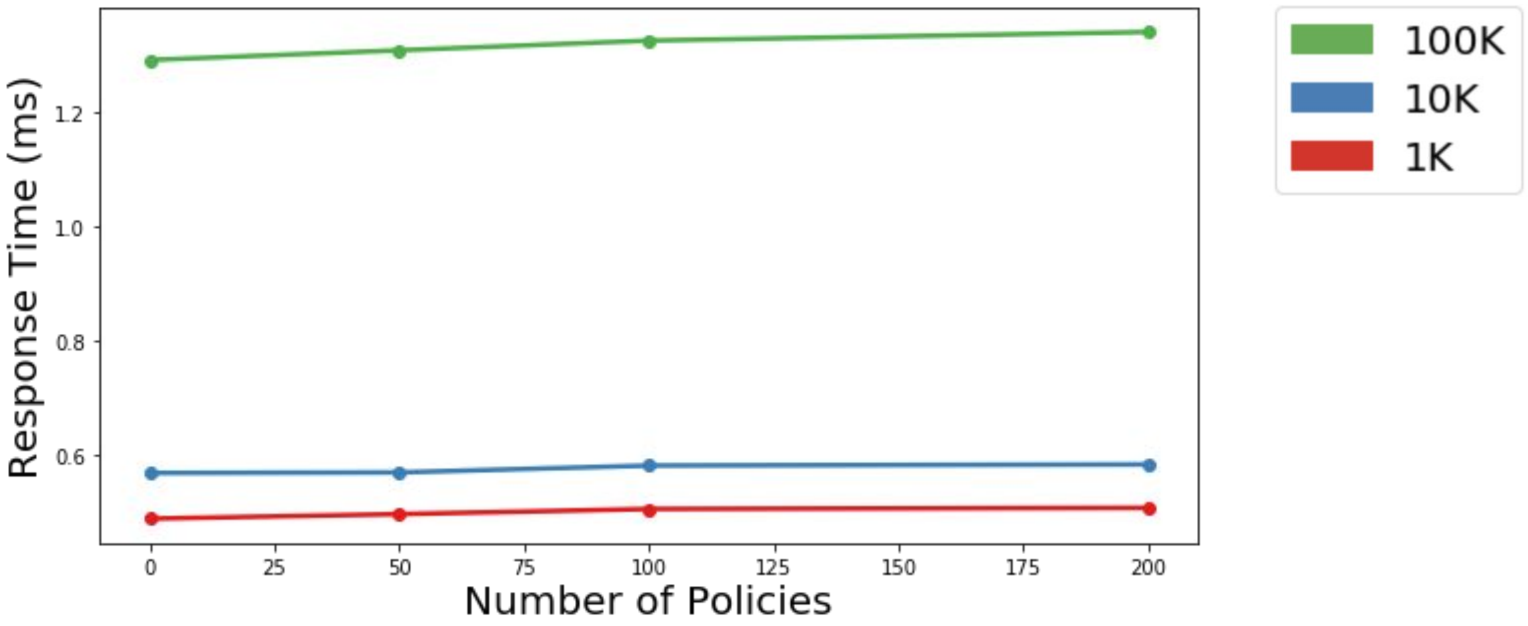


- Varied network policy from 0 to 200 policies

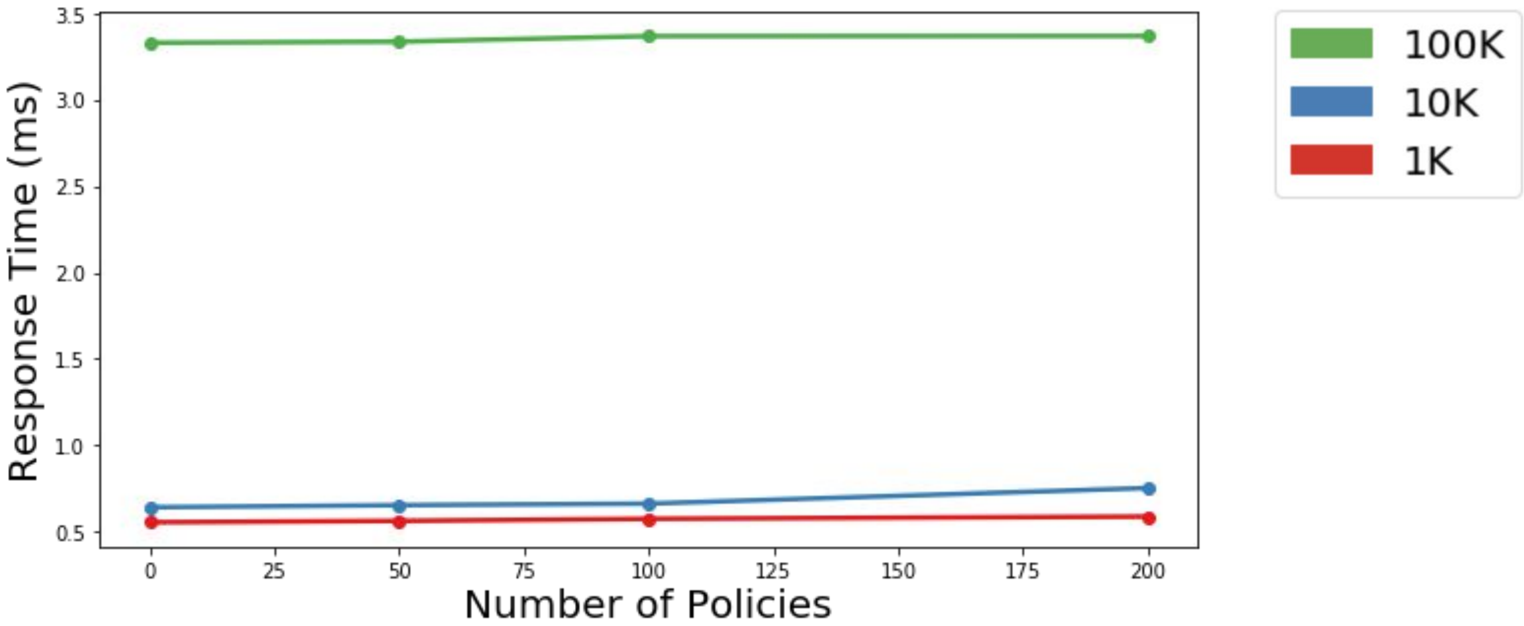
Network Policy (Calico)



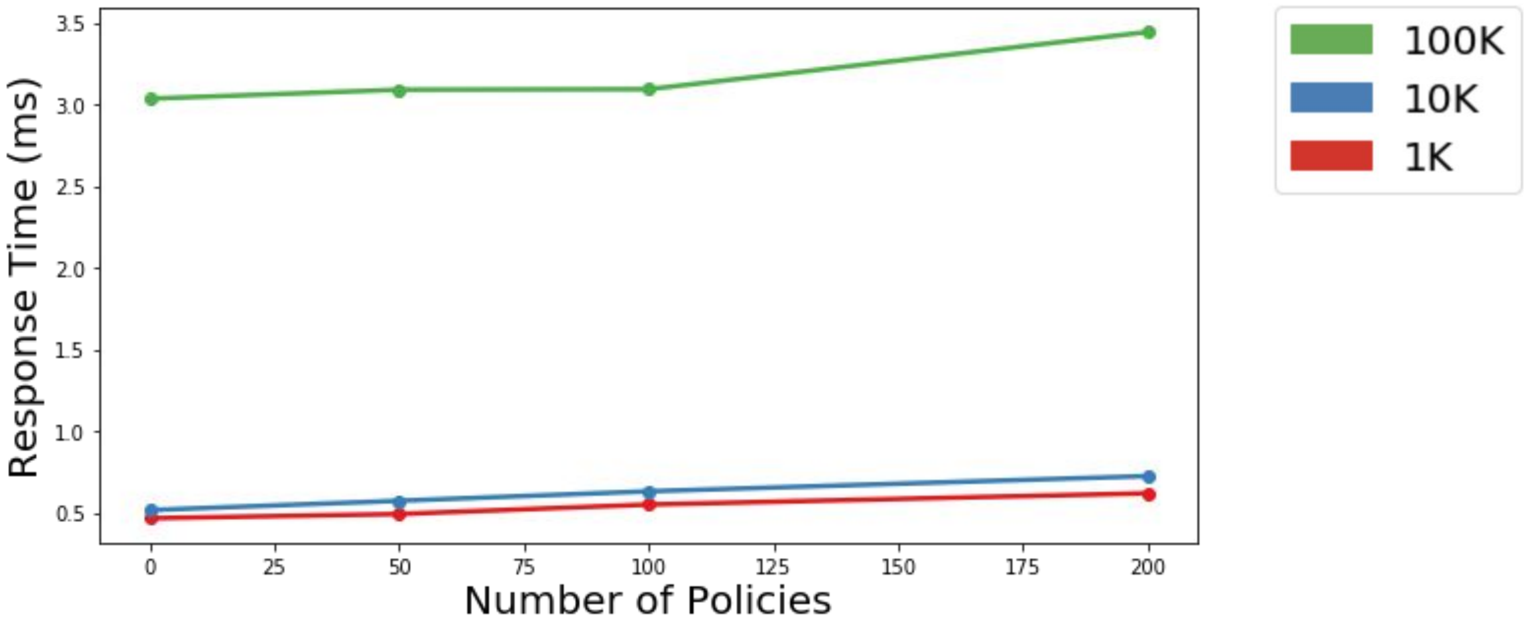
Network Policy (Weave)



Network Policy (Cilium)



Network Policy (Kube-Router)



KubeCon | CloudNativeCon

China 2018

Takeaways

Takeaways

CNI

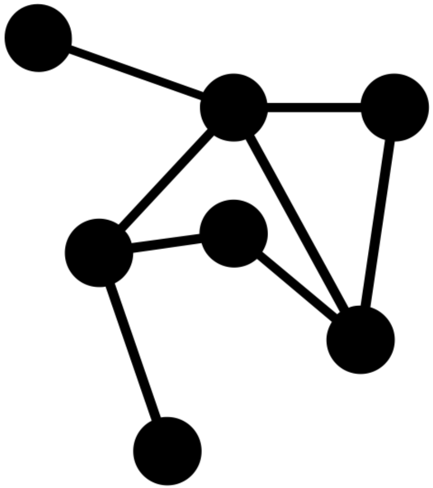
Very challenging to pick appropriate CNI plugins

Tradeoffs between **performance, security, isolation**

Takeaways

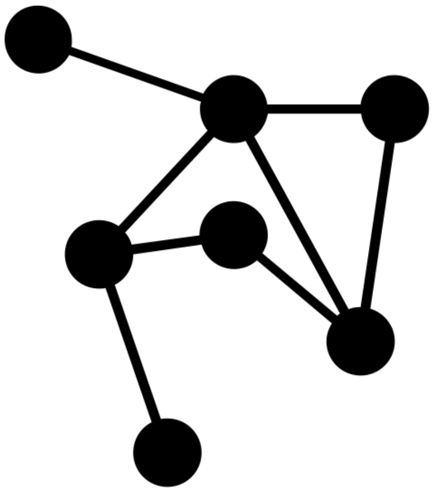
In general, with default config,
Flannel, Weave, Romana achieves better network performance than the others

Takeaways



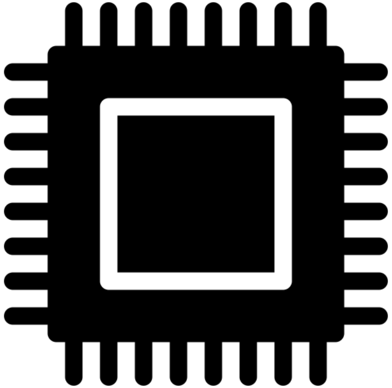
Most CNI plugins get much **larger throughput** to **TCP** than **UDP** workloads

Takeaways



Most CNI plugins **scaled more** in **UDP**
with **considerable loss**

Takeaways



Most CNI plugins introduce **CPU overhead** with **Weave** and **Cilium** being largest

Takeaways

Most CNI plugins add **reasonable delay** to **launch containers**, except **Flannel**

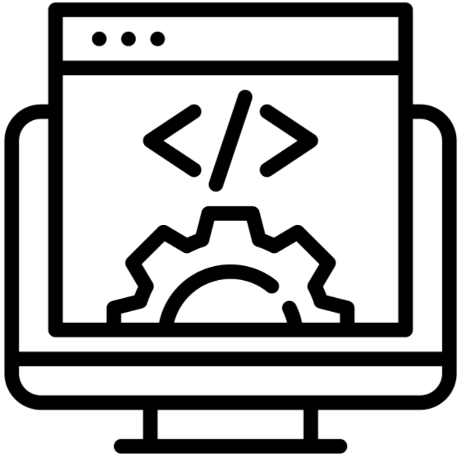
Takeaways

Most CNI plugins introduce **small delay** as number of **policies increases**

Next

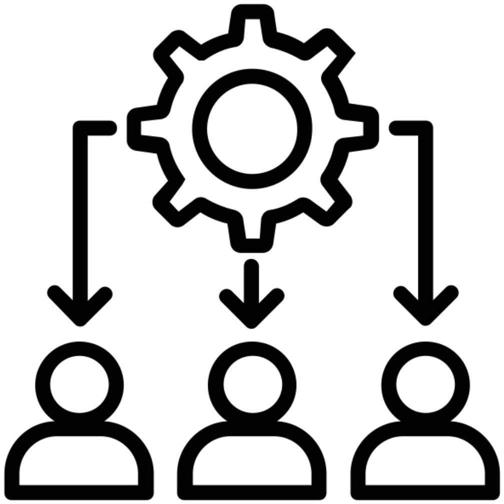
Need more **comprehensive analysis**
on the experiment results

Next



Perform specific **configuration**
to allow **better performance** on some CNI Plugins

Next



More types of experiments
on **concurrent requests** and **workload interference**

References

An Analysis and Empirical Study of Container Networks, Kun Suo et.al
http://ranger.uta.edu/~jrao/papers/INFOCOM18.pdf

Comparison of Networking Solutions for Kubernetes, Machine Zone
http://machinezone.github.io/research/networking-solutions-for-kubernetes

High Performance Network Policies in Kubernetes Clusters, Pani Networks
https://kubernetes.io/blog/2016/09/high-performance-network-policies-kubernetes

Acknowledgements

- Ajey Gore, GO-JEK
- Gaetano Borgione, VMware
- Ali Khayam, Amazon Web Services
- Samudra Bekti, Cisco

KubeCon | CloudNativeCon

China 2018

谢谢

Giri Kuncoro, giri.kuncoro@go-jek.com
Vijay Dhama, vijay.dhama@go-jek.com