KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

Alcor

Hyperscale Cloud Network Management

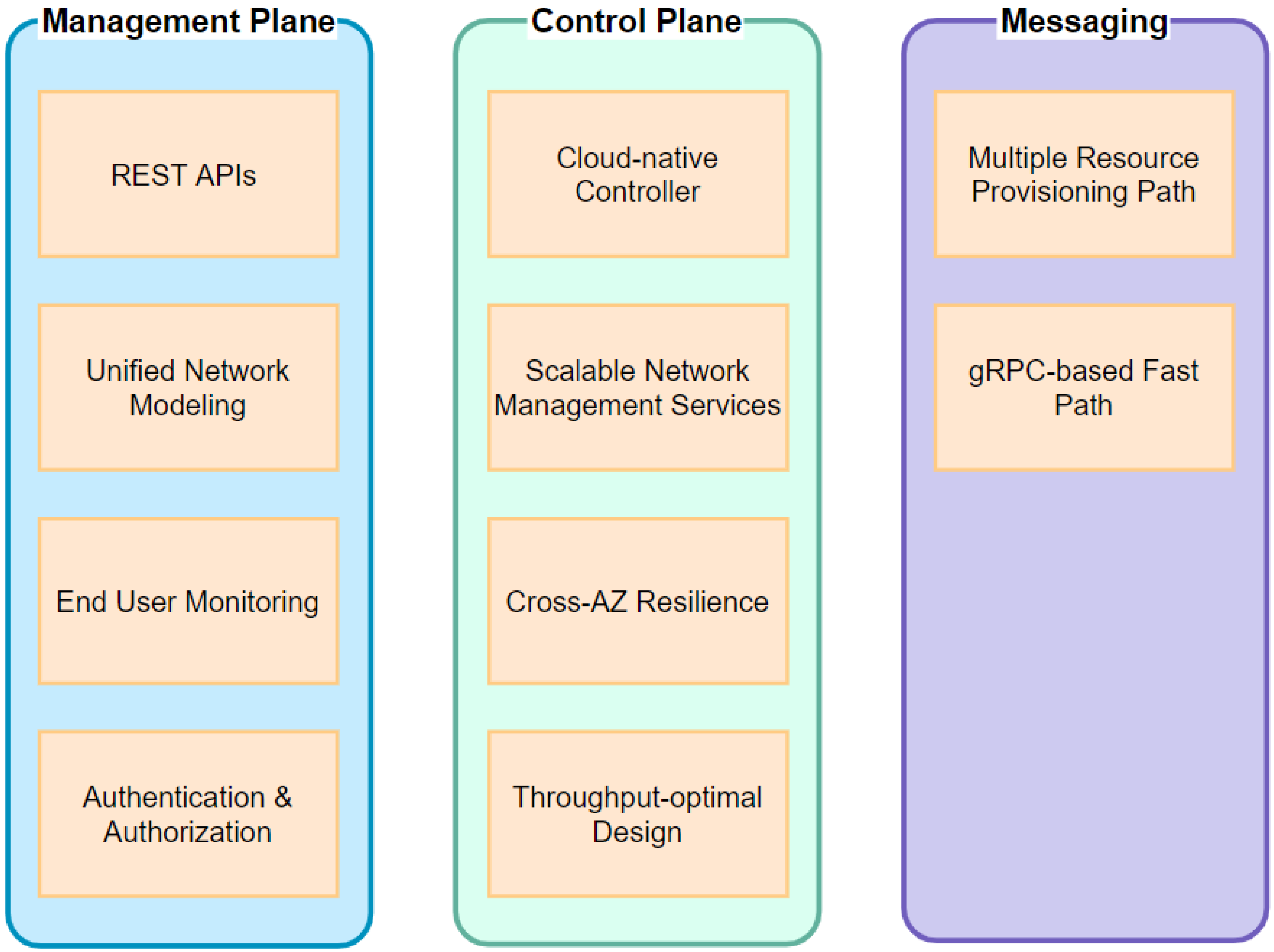
Futurewei Technologies

Introduction

- ➢ Alcor is a cloud native SDN platform powered by Kubernetes/Istio

P Performance	A Availability	S Scalability	E Extensibility
• **Throughput-optimal design** to allow batched provisioning of network resources • **Fast provisioning path** to support time-critical applications such as serviceless	• **Always-on control plane** without a single point of failure • **Cross-AZ resilience** for services and data • **Fault-tolerant design** with multiple resource provisioning paths	• **Large-scale network resource management** • Scale to half a million hosts and tens of millions network ports	• **Unified resource management** of both VMs and containers • **Plugable model** to support various implementations of data plane

Architecture Highlight



Management Plane

- REST APIs
- Unified Network Modeling
- End User Monitoring
- Authentication & Authorization

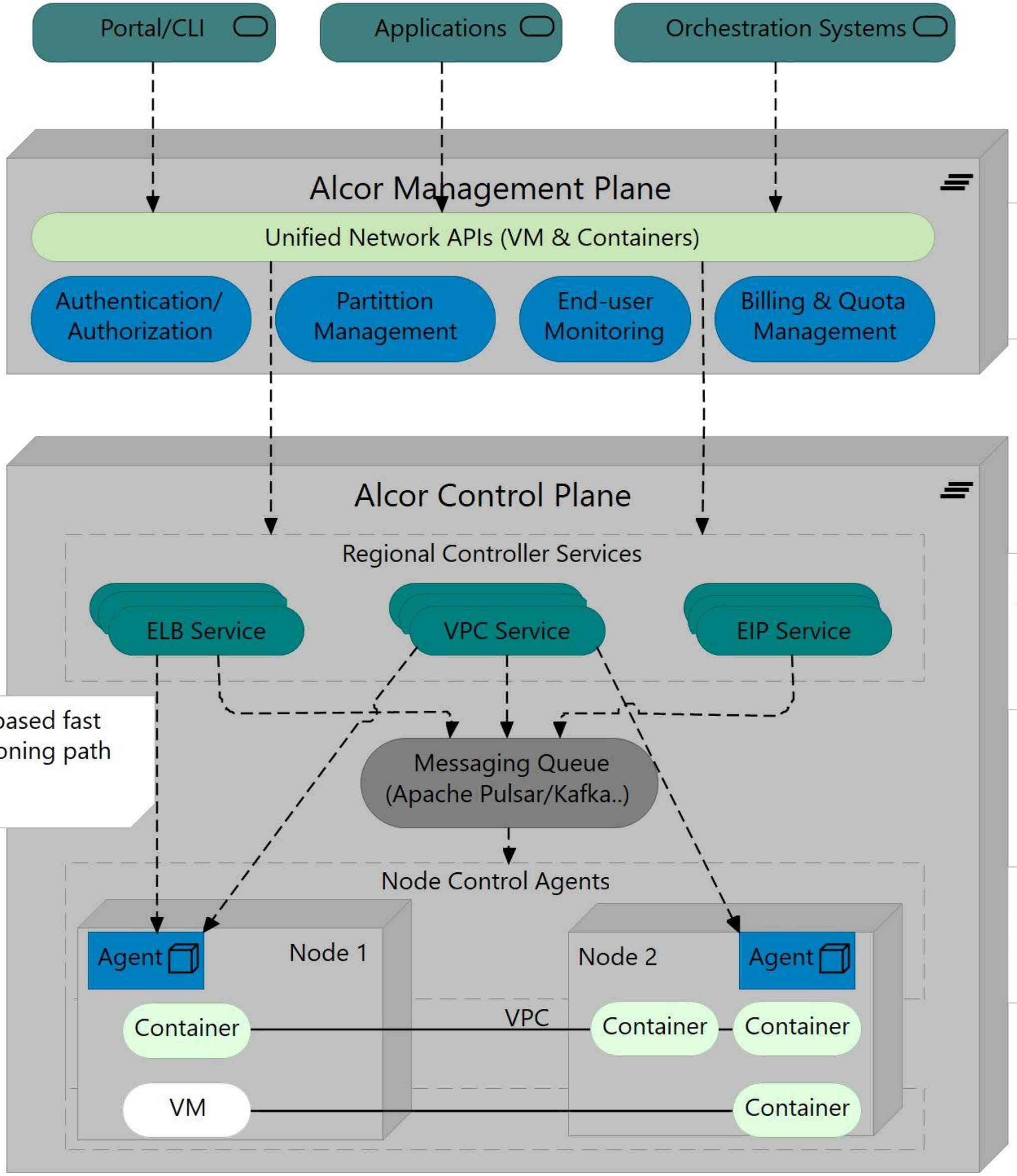
Control Plane

- Cloud-native Controller
- Scalable Network Management Services
- Cross-AZ Resilience
- Throughput-optimal Design

Messaging

- Multiple Resource Provisioning Path
- gRPC-based Fast Path

Architecture Overview



Portal/CLI

Applications

Orchestration Systems

Alcor Management Plane

Unified Network APIs (VM & Containers)

Authentication/ Authorization

Partittion Management

End-user Monitoring

Billing & Quota Management

Alcor Control Plane

Regional Controller Services

ELB Service

VPC Service

EIP Service

Messaging Queue (Apache Pulsar/Kafka..)

Node Control Agents

Agent

Node 1

Node 2

Agent

Container

VPC

Container

Container

VM

Container

Rest APIs for unified network management of both containers and VMs

Distributed micro-services architecture to support low provisioning latency and high throughput for hyper-scale cloud deployments

gRPC-based fast provisioning path

Per-host stateless agent programs data plane with resilience, diagnosability, monitoring and compatibility

Cloud-Native Control Plane

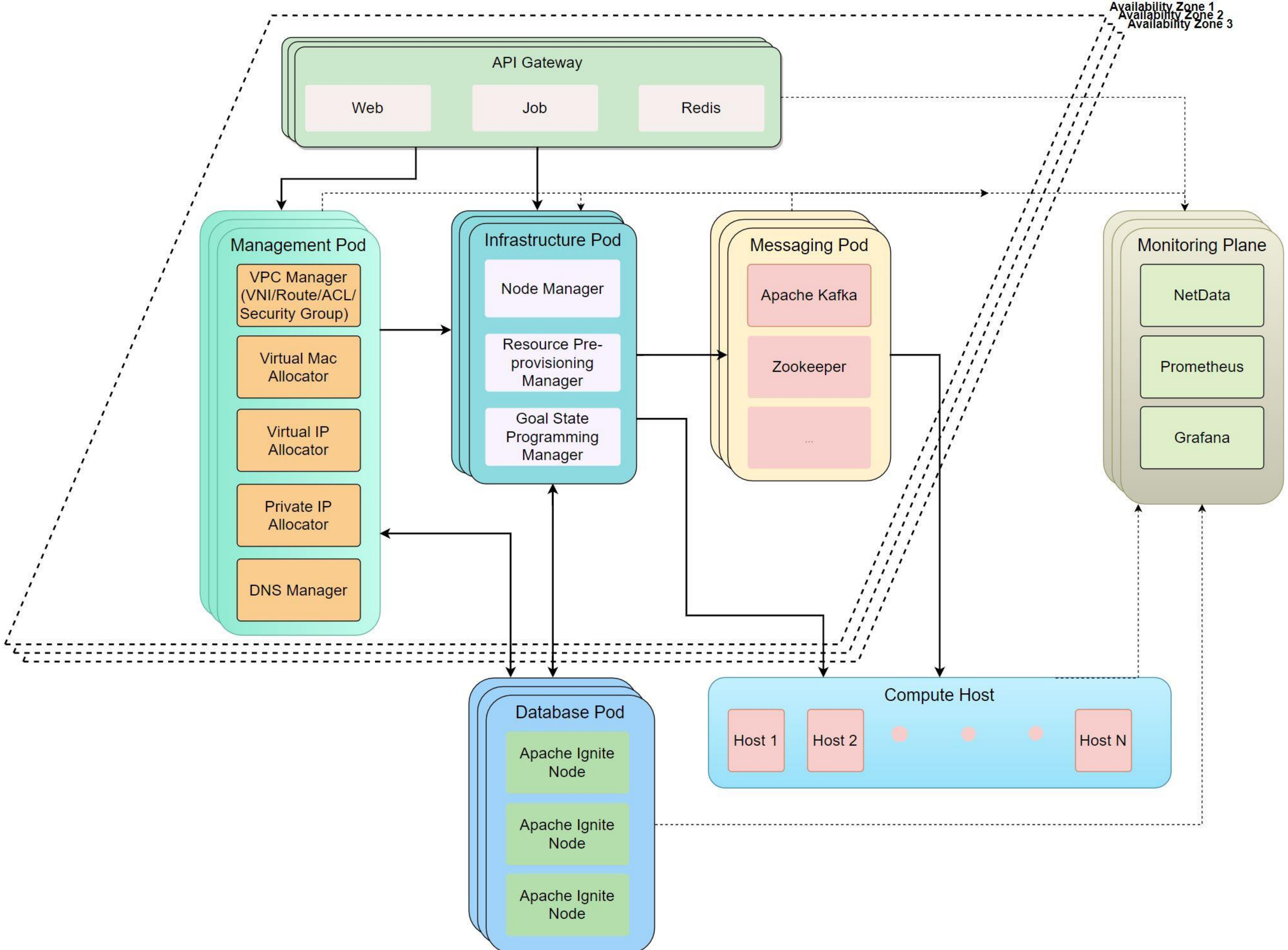
KubeCon | CloudNativeCon North America 2019

Cloud-native application powered by Kubernetes

- Each controller instance is a Kubernetes app
- Each app contains multiple services

Micro-Services Architecture

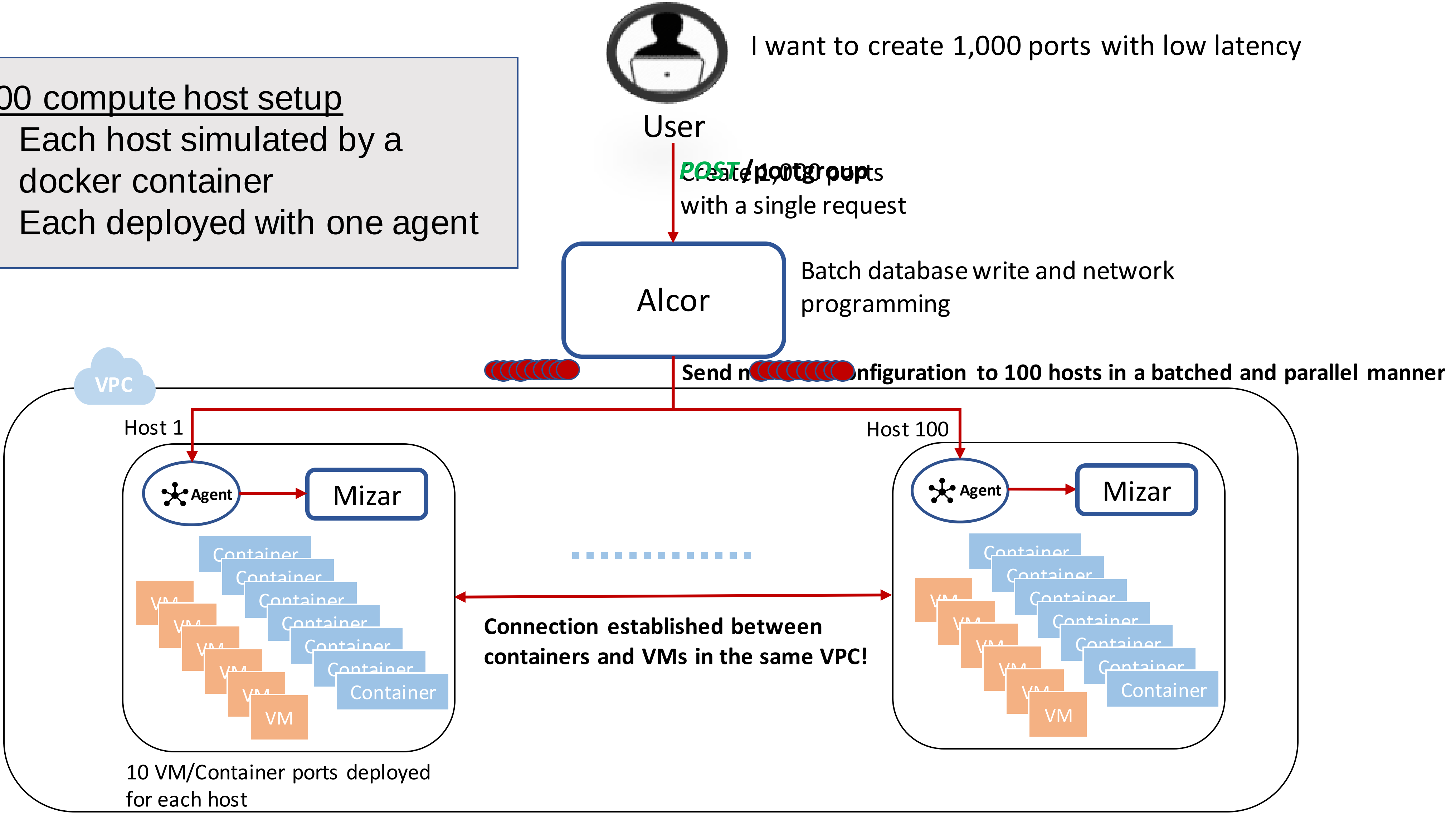
- Secure, connect, and monitor control plane micro-services with Istio
- Fine-grained control of service-to-service communication including load balancing, retries, failovers, and rate limits.



User Scenario: Large-Scale VPC Provisioning

100 compute host setup

- ○ Each host simulated by a docker container
- ○ Each deployed with one agent



I want to create 1,000 ports with low latency

POST /portgroup

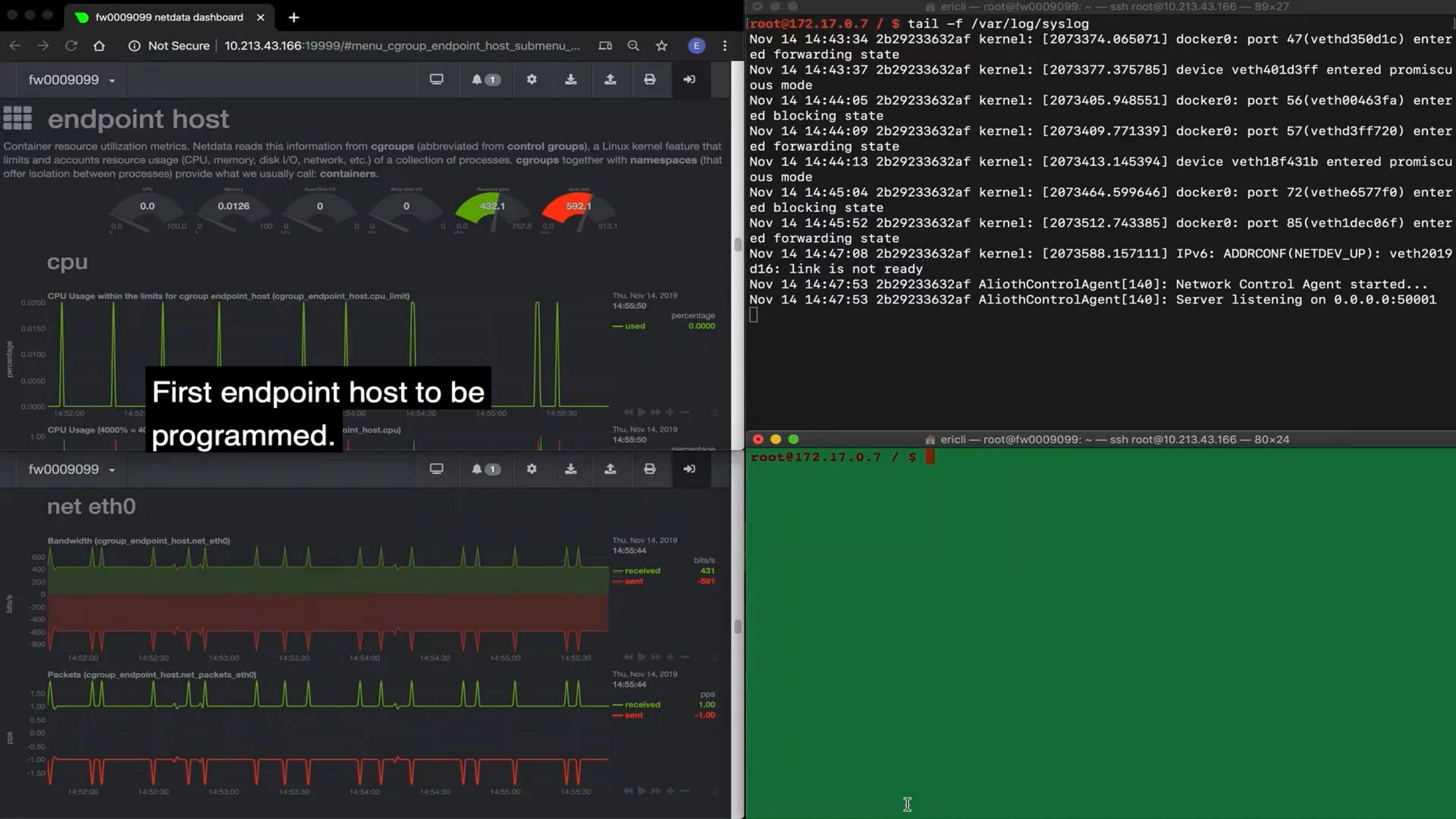
Create 1,000 ports with a single request

Batch database write and network programming

Send n[illegible]nfiguration to 100 hosts in a batched and parallel manner

Connection established between containers and VMs in the same VPC!

10 VM/Container ports deployed for each host



fw0009099 netdata dashboard
Not Secure | 10.213.43.166:19999/#menu_cgroup_endpoint_host_submenu_...

endpoint host

Container resource utilization metrics. Netdata reads this information from **cgroups** (abbreviated from **control groups**), a Linux kernel feature that limits and accounts resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes. **cgroups** together with **namespaces** (that offer isolation between processes) provide what we usually call: **containers**.

cpu

CPU Usage within the limits for cgroup endpoint_host (cgroup_endpoint_host.cpu_limit)
Thu, Nov 14, 2019
14:55:50
percentage
used 0.0000
0.0200
0.0150
0.0100
0.0050
0.0000
14:52:00
14:54:30
14:55:00
14:55:30
CPU Usage (4000% = 40 ... oint_host.cpu)
1.00

First endpoint host to be programmed.

net eth0

```
root@172.17.0.7 / $ tail -f /var/log/syslog
Nov 14 14:43:34 2b29233632af kernel: [2073374.065071] docker0: port 47(vethd350d1c) entered forwarding state
Nov 14 14:43:37 2b29233632af kernel: [2073377.375785] device veth401d3ff entered promiscuous mode
Nov 14 14:44:05 2b29233632af kernel: [2073405.948551] docker0: port 56(veth00463fa) entered blocking state
Nov 14 14:44:09 2b29233632af kernel: [2073409.771339] docker0: port 57(vethd3ff720) entered forwarding state
Nov 14 14:44:13 2b29233632af kernel: [2073413.145394] device veth18f431b entered promiscuous mode
Nov 14 14:45:04 2b29233632af kernel: [2073464.599646] docker0: port 72(vethe6577f0) entered blocking state
Nov 14 14:45:52 2b29233632af kernel: [2073512.743385] docker0: port 85(veth1dec06f) entered forwarding state
Nov 14 14:47:08 2b29233632af kernel: [2073588.157111] IPv6: ADDRCONF(NETDEV_UP): veth2019d16: link is not ready
Nov 14 14:47:53 2b29233632af AliothControlAgent[140]: Network Control Agent started...
Nov 14 14:47:53 2b29233632af AliothControlAgent[140]: Server listening on 0.0.0.0:50001
```

```
root@172.17.0.7 / $ 
```

Throughput-Optimal Design

Focus on throughout optimization on every system layer

API

- Group of ports deployment with one POST call
- Unified network resource management for both VMs and containers

Controller

- Implicit batching for database write and network programming
- Per-host network configuration batching

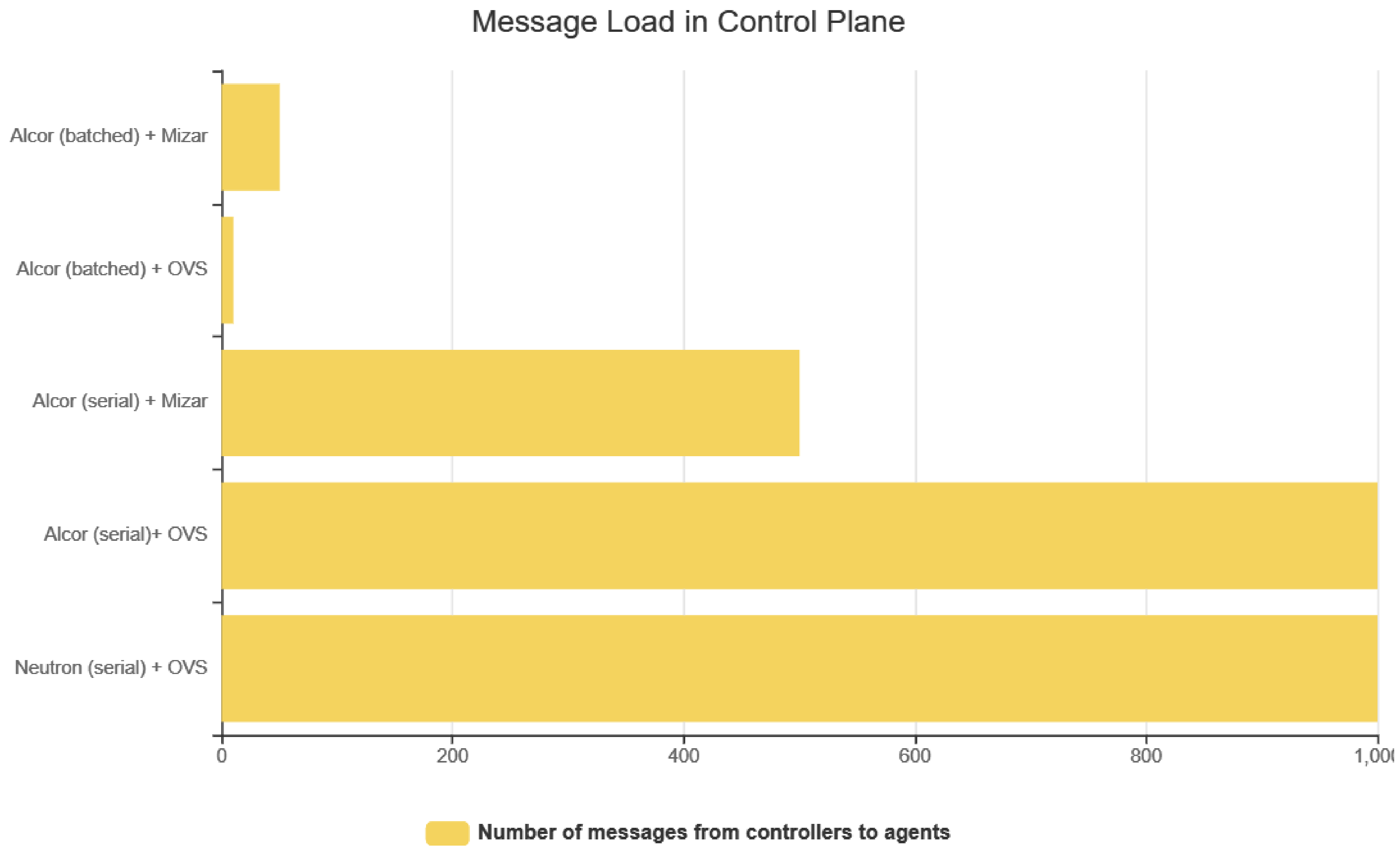
Messaging

- Drive groups of resources to the same host in one shot
- Support various combinations of resource updates
  - Multiple resource instances
  - Multiple resource types
  - Across VPC/subnet boundaries

Host Agent

- Parallel network setup on the host and port programming to data plane
- Achieve 1000+ port RPM on the host with Mizar data plane

Message Load in Control Plane



Batch vs Serial comm.

90% to 99% cut on message count when compared with serial communication

on average 50% cut on message load

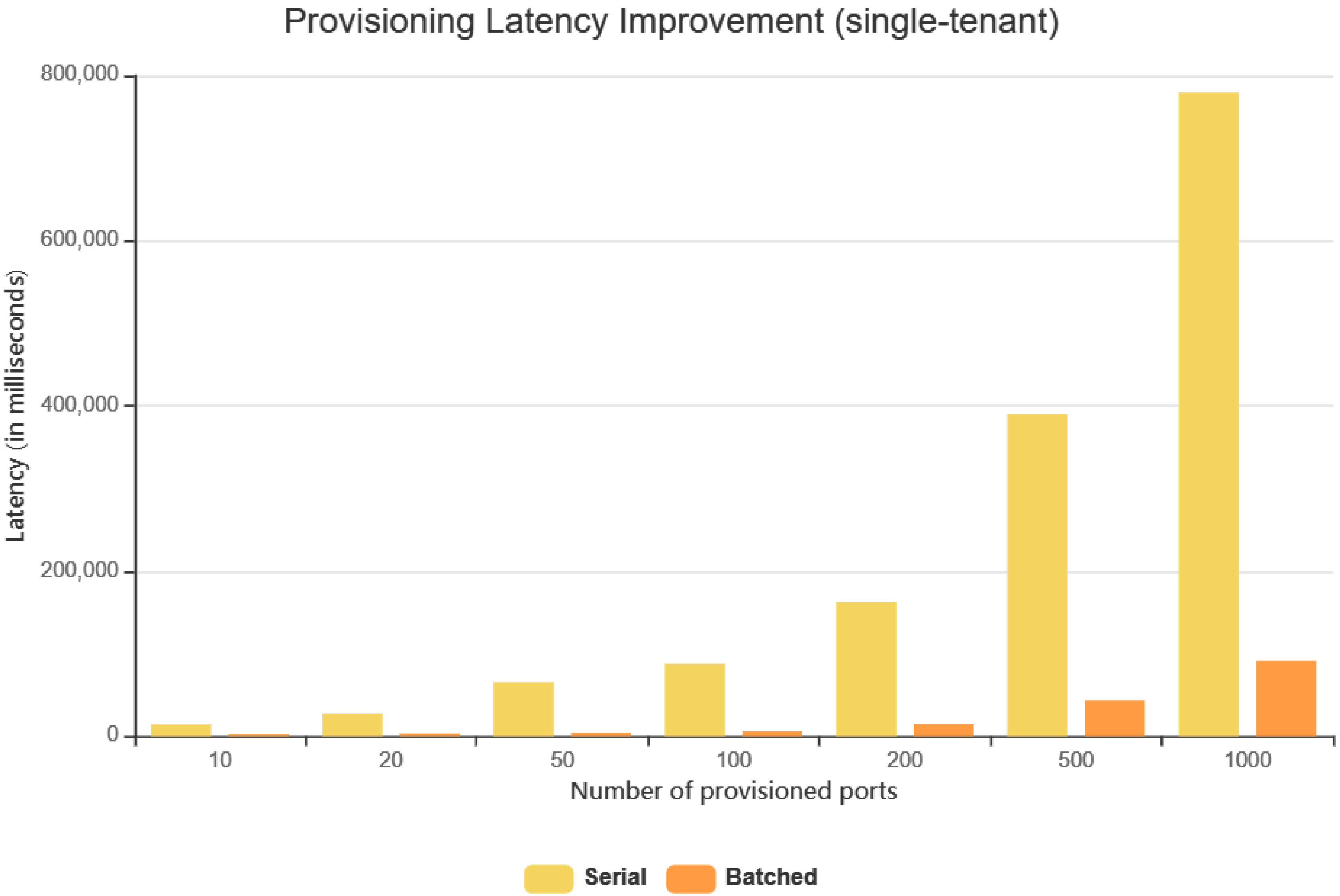
E2E Provisioning Latency

Batch vs. Serial Provisioning

- 88% to 95% latency reduction for large deployments
- Complete 1000 ports programming within 95 seconds

Batch Provisioning Process

- Created with a single API call
- Distributed to hosts in a batched and parallel manner



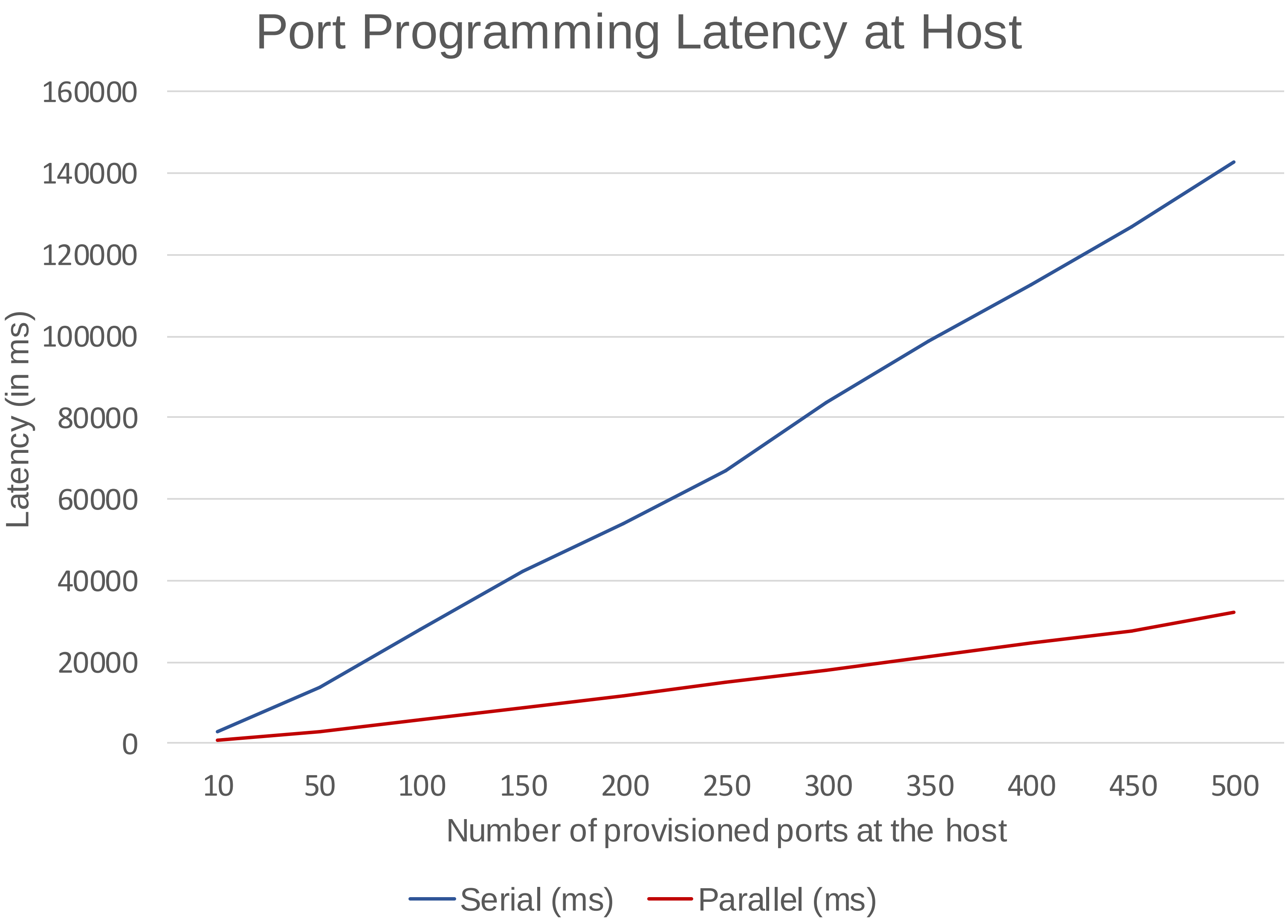
Agent Programming Latency

Batch vs. Serial Host Programming

- 78% programming latency reduction on the hosts
- Scale to 500 ports per host within 33 seconds

Parallel Host Programming Process

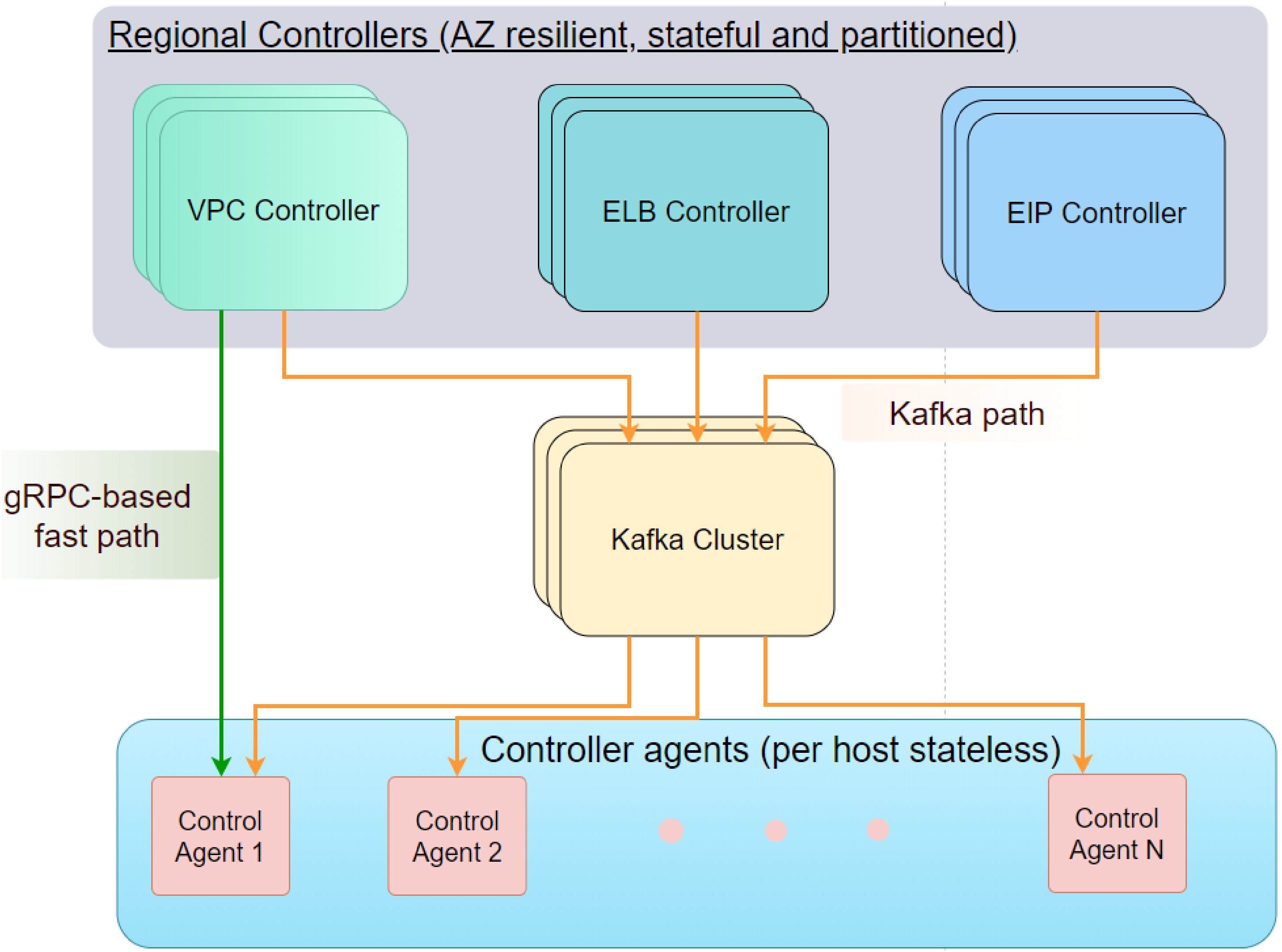
- Multiple threading for network configuration
- Program data plane in a single-threaded mode or multi-threaded mode

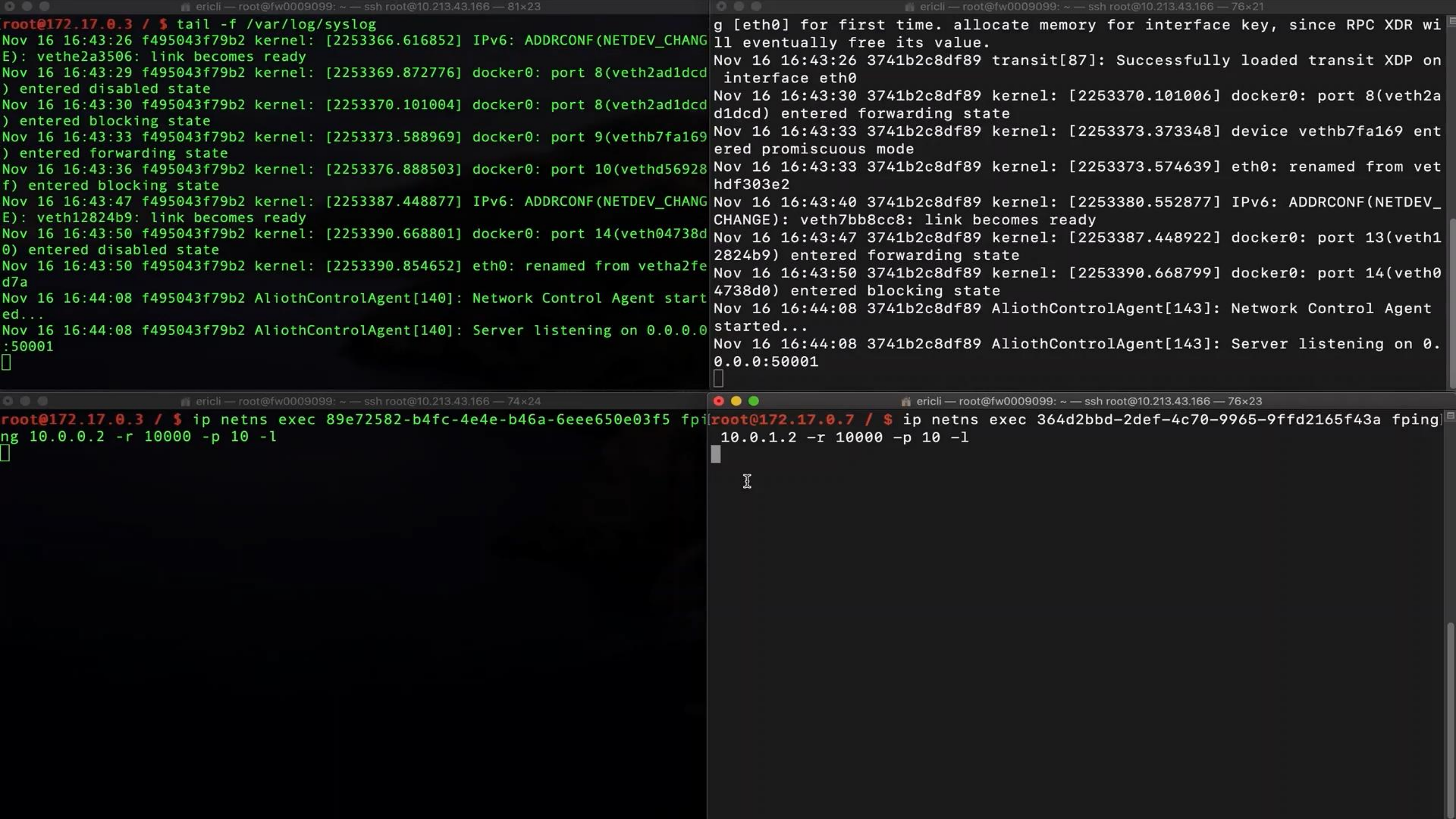


Fast Port Provisioning

Use Scenarios: Time-critical application

- ➢ Direct communication channel from controller to agent
- ➢ Alternative provisioning path for control plane reliability





ericli — root@fw0009099: ~ — ssh root@10.213.43.166 — 81×23

```
root@172.17.0.3 / $ tail -f /var/log/syslog
Nov 16 16:43:26 f495043f79b2 kernel: [2253366.616852] IPv6: ADDRCONF(NETDEV_CHANG
E): vethe2a3506: link becomes ready
Nov 16 16:43:29 f495043f79b2 kernel: [2253369.872776] docker0: port 8(veth2ad1dcd
) entered disabled state
Nov 16 16:43:30 f495043f79b2 kernel: [2253370.101004] docker0: port 8(veth2ad1dcd
) entered blocking state
Nov 16 16:43:33 f495043f79b2 kernel: [2253373.588969] docker0: port 9(vethb7fa169
) entered forwarding state
Nov 16 16:43:36 f495043f79b2 kernel: [2253376.888503] docker0: port 10(vethd56928
f) entered blocking state
Nov 16 16:43:47 f495043f79b2 kernel: [2253387.448877] IPv6: ADDRCONF(NETDEV_CHANG
E): veth12824b9: link becomes ready
Nov 16 16:43:50 f495043f79b2 kernel: [2253390.668801] docker0: port 14(veth04738d
0) entered disabled state
Nov 16 16:43:50 f495043f79b2 kernel: [2253390.854652] eth0: renamed from vetha2fe
d7a
Nov 16 16:44:08 f495043f79b2 AliothControlAgent[140]: Network Control Agent start
ed...
Nov 16 16:44:08 f495043f79b2 AliothControlAgent[140]: Server listening on 0.0.0.0
:50001
```

ericli — root@fw0009099: ~ — ssh root@10.213.43.166 — 76×21

```
g [eth0] for first time. allocate memory for interface key, since RPC XDR wi
ll eventually free its value.
Nov 16 16:43:26 3741b2c8df89 transit[87]: Successfully loaded transit XDP on
 interface eth0
Nov 16 16:43:30 3741b2c8df89 kernel: [2253370.101006] docker0: port 8(veth2a
d1dcd) entered forwarding state
Nov 16 16:43:33 3741b2c8df89 kernel: [2253373.373348] device vethb7fa169 ent
ered promiscuous mode
Nov 16 16:43:33 3741b2c8df89 kernel: [2253373.574639] eth0: renamed from vet
hdf303e2
Nov 16 16:43:40 3741b2c8df89 kernel: [2253380.552877] IPv6: ADDRCONF(NETDEV_
CHANGE): veth7bb8cc8: link becomes ready
Nov 16 16:43:47 3741b2c8df89 kernel: [2253387.448922] docker0: port 13(veth1
2824b9) entered forwarding state
Nov 16 16:43:50 3741b2c8df89 kernel: [2253390.668799] docker0: port 14(veth0
4738d0) entered blocking state
Nov 16 16:44:08 3741b2c8df89 AliothControlAgent[143]: Network Control Agent
started...
Nov 16 16:44:08 3741b2c8df89 AliothControlAgent[143]: Server listening on 0.
0.0.0:50001
```

ericli — root@fw0009099: ~ — ssh root@10.213.43.166 — 74×24

```
root@172.17.0.3 / $ ip netns exec 89e72582-b4fc-4e4e-b46a-6eee650e03f5 fpi
ng 10.0.0.2 -r 10000 -p 10 -l
```

ericli — root@fw0009099: ~ — ssh root@10.213.43.166 — 76×23

```
root@172.17.0.7 / $ ip netns exec 364d2bbd-2def-4c70-9965-9ffd2165f43a fping
 10.0.1.2 -r 10000 -p 10 -l
```

Thank you!

Contact

Liguang Xie (lxie@futurewei.com), Ying Xiong (yxiong@futurewei.com)
Seattle Cloud Lab
Futurewei Technologies