Zeta: A Scalable and Robust East-West Communication Framework in Large-Scale Clouds

Speaker: Zhuolong Yu (Johns Hopkins University)

Authors: Qianyu Zhang¹, Gongming Zhao¹, Hongli Xu¹, Zhuolong Yu², Liguang Xie³, Yangming Zhao¹, Chunming Qiao⁴, Ying Xiong³, Liusheng Huang¹
Cloud Network Virtualization

- **Cloud Virtual Network**
  - Brings great benefits (e.g., flexibility, scalability)
  - Requires *high-speed* and *reliable* connectivity
Cloud Network Virtualization

- **Cloud Virtual Network**
  - Brings great benefits (e.g., flexibility, scalability)
  - Requires *high-speed* and *reliable* connectivity

- **East-West Communication (Intra-Datacenter Communication)**
  - Communication between instances within a data center
  - Contributes to a large portion of network traffics in cloud (e.g., 75%)
Cloud Network Virtualization

• Large-Scale
  • Over 100k servers and millions of instances

• Dynamics
  • Frequent launching and short life cycles of instances

Everything at Google runs in containers

Containerization helps our development teams move fast, deploy software efficiently, and operate at an unprecedented scale. We've packaged over a decade's worth of experience launching several billion containers per week into Google Cloud so that developers and businesses of any size can easily tap the latest in container innovation.

Learn more about containers and their benefits
Challenges

Scalability

Numerous forwarding rules introduce great overhead

Robustness

Device failures and burst traffic can lead to network congestion/interruption

Low-latency

The latency of establishing communication is critical to service deployment
Existing Communication Models

• **Preprogrammed Model**
  • Pre-install all potential rules
  • Direct path with low latency
  • E.g., NVP (NSDI’2014)

• **Limitations**
  • Poor scalability
  • High VMs launching/migration latency
Existing Communication Models

- **On-Demand Model**
  - 1\textsuperscript{st} packet of a flow is sent to the controller
  - Controller programs the forwarding rules
  - Better scalability

- **Limitations**
  - 1\textsuperscript{st} packet has high latency
  - Highly relies on controller
Existing Communication Models

• **Gateway Model**
  • Default rules forwarding traffic to the gateway
  • Gateways offloads elephant flows
  • E.g., Andromeda (NSDI’2018)

• **Limitations**
  • High recovery latency from gateway failure
  • Load imbalance and overloaded under burst traffic
### Existing Communication Models

<table>
<thead>
<tr>
<th>Models</th>
<th>Robustness</th>
<th>Scalability</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprogrammed</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>On-Demand</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Gateway</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Zeta</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Zeta Overview

Framework Management Layer

Gateway Cluster Layer

Gateway Cluster 1
- Node 1
- Node 2

Controller (K8S)

Gateway Cluster N
- Node 1
- Node 2
- Node 3

On-host Forwarding Layer

Host 1
- Zeta Agent
- OVS
- VM
- VM
- VM

Host 2
- OVS
- Zeta Agent
- VM
- VM
- VM

① Header Packets
② OAM Packet
③ Direct Path
Zeta Overview

Gateway cluster and group table for load balancing
Zeta Overview

XDP-based forwarding, detection, and offloading

Gateway cluster and group table for load balancing
Zeta Overview

Framework Management Layer

Gateway Cluster Layer

Gateway Cluster 1
  - Node 1
  - Node 2

Controller (K8S)

Gateway Cluster N
  - Node 1
  - Node 2
  - Node 3

On-host Forwarding Layer

Zeta Agent
  - OVS

Host 1
  - VM
  - VM
  - VM

OVS

Host 2
  - VM
  - VM
  - VM

Zeta Agent

① Header Packets

② OAM Packet

③ Direct Path

Multi-IPs for fast recovery

XDP-based forwarding, detection, and offloading

Gateway cluster and group table for load balancing

VM

Zeta Overview

XDP-based forwarding, detection, and offloading

Multi-IPs for fast recovery

Gateway cluster and group table for load balancing
Zeta Overview

Higher level management and load adjustment

Multi-IPs for fast recovery

XDP-based forwarding, detection, and offloading

Gateway cluster and group table for load balancing

Zeta Agent

OVS

Controller (K8S)

OVS

VM

VM

VM

VM

VM

VM

VM

Framework Management Layer

Gateway Cluster Layer

Node 1

Node 2

Node 1

Node 2

Node 3

On-host Forwarding Layer

① Header Packets

② OAM Packet

③ Direct Path

Zeta Agent

Host 1

Host 2

Multi-IPs for fast recovery

Higher level management and load adjustment

XDP-based forwarding, detection, and offloading

Gateway cluster and group table for load balancing
**On-host Forwarding: Load Balancing with Group Table**

### Flow Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC_id == 1</td>
<td>Gid := 1</td>
</tr>
<tr>
<td>VPC_id == 2</td>
<td>Gid := 2</td>
</tr>
</tbody>
</table>

### Group Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td>Gid == 2</td>
<td>Dst := ip3</td>
</tr>
<tr>
<td></td>
<td>Dst := ip4</td>
</tr>
</tbody>
</table>
On-host Forwarding: Load Balancing with Group Table

Flow Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC_id == 1</td>
<td>Gid := 1</td>
</tr>
<tr>
<td>VPC_id == 2</td>
<td>Gid := 2</td>
</tr>
</tbody>
</table>

Group Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td>Gid == 2</td>
<td>Dst := ip3</td>
</tr>
<tr>
<td></td>
<td>Dst := ip4</td>
</tr>
</tbody>
</table>
On-host Forwarding: Load Balancing with Group Table

Flow Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC_id == 1</td>
<td>Gid := 1</td>
</tr>
<tr>
<td>VPC_id == 2</td>
<td>Gid := 2</td>
</tr>
</tbody>
</table>

Group Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td>Gid == 2</td>
<td>Dst := ip3</td>
</tr>
<tr>
<td></td>
<td>Dst := ip4</td>
</tr>
</tbody>
</table>

GW 1:1
ip1
GW 1:2
ip2
GW 2:1
ip3
GW 2:2
ip4

Cluster 1

Cluster 2

ToR Switch

VPC 1

VPC 2

VM 1

VM 2

VM 3

VM 4
On-host Forwarding: Load Balancing with Group Table

Flow Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC_id == 1</td>
<td>Gid := 1</td>
</tr>
<tr>
<td>VPC_id == 2</td>
<td>Gid := 2</td>
</tr>
</tbody>
</table>

Group Table:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td>Gid == 2</td>
<td>Dst := ip3</td>
</tr>
<tr>
<td></td>
<td>Dst := ip4</td>
</tr>
</tbody>
</table>
Gateway Cluster: XDP-Based Forwarding

- **Forwarding Module**
  - Forwards traffic
  - Detects elephant flows
  - Offloads direct path rules

- **Storage Module**
  - Consists of several eBPF maps
  - Stores forwarding rules
  - Real-time flow information
Gateway Cluster: XDP-Based Forwarding

**Forwarding Module**
- Forwards traffic
- Detects elephant flows
- Offloads direct path rules

**Storage Module**
- Consists of several eBPF maps
- Stores forwarding rules
- Real-time flow information
Gateway Cluster: Fast Recovery with Multi IPs

ToR Switch

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td></td>
<td>Dst := ip3</td>
</tr>
</tbody>
</table>

Node 1: ip1
Node 2: ip2
Node 3: ip3

vSwitch
VM VM...

vSwitch
VM VM...

vSwitch
VM VM...
Gateway Cluster: Fast Recovery with Multi IPs

Node 1
ip1

Node 2

Node 3
ip3

ToR Switch

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gid == 1</td>
<td>Dst := ip1</td>
</tr>
<tr>
<td></td>
<td>Dst := ip2</td>
</tr>
<tr>
<td></td>
<td>Dst := ip3</td>
</tr>
</tbody>
</table>
Gateway Cluster: Fast Recovery with Multi IPs

Node 1 (ip1) → vSwitch → VM → vSwitch → VM → vSwitch → VM

Node 2 (X) → vSwitch → VM → vSwitch → VM → vSwitch → VM

Node 3 (ip3) → vSwitch → VM → vSwitch → VM → vSwitch → VM

ToR Switch

Match | Action
--- | ---
Gid == 1 | Dst := ip1
Dst := ip2
Dst := ip3

…
Gateway Cluster: Fast Recovery with Multi IPs

Node 1
ip1
Node 2
ip3
Node 3
ip5

ToR Switch

Multi IPs Mapping Table:

<table>
<thead>
<tr>
<th>Node</th>
<th>vIP</th>
<th>vMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node1</td>
<td>ip1</td>
<td>mac1</td>
</tr>
<tr>
<td>Node2</td>
<td>ip3</td>
<td>mac3</td>
</tr>
<tr>
<td>Node3</td>
<td>ip5</td>
<td>mac5</td>
</tr>
</tbody>
</table>

Match | Action
---|---
mac3 | port2
mac4 | port2

vSwitch
VM
VM

vSwitch
VM
VM

vSwitch
VM
VM

......
Gateway Cluster: Fast Recovery with Multi IPs

Multi IPs Mapping Table:

<table>
<thead>
<tr>
<th>Node</th>
<th>vIP</th>
<th>vMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node1</td>
<td>ip1</td>
<td>mac1</td>
</tr>
<tr>
<td></td>
<td>ip2</td>
<td>mac2</td>
</tr>
<tr>
<td>Node2</td>
<td>ip3</td>
<td>mac3</td>
</tr>
<tr>
<td></td>
<td>ip4</td>
<td>mac3</td>
</tr>
<tr>
<td>Node3</td>
<td>ip5</td>
<td>mac5</td>
</tr>
<tr>
<td></td>
<td>ip6</td>
<td>mac6</td>
</tr>
</tbody>
</table>

Match | Action
------|-------
mac3  | port2
mac4  | port2

ToR Switch

vSwitch

VM VM

......
Gateway Cluster: Fast Recovery with Multi IPs

Multi IPs Mapping Table:

<table>
<thead>
<tr>
<th>Node</th>
<th>vIP</th>
<th>vMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node1</td>
<td>ip1</td>
<td>mac1</td>
</tr>
<tr>
<td></td>
<td>ip2</td>
<td>mac2</td>
</tr>
<tr>
<td>Node2</td>
<td>ip3</td>
<td>mac3</td>
</tr>
<tr>
<td></td>
<td>ip4</td>
<td>mac4</td>
</tr>
<tr>
<td>Node3</td>
<td>ip5</td>
<td>mac5</td>
</tr>
<tr>
<td></td>
<td>ip6</td>
<td>mac6</td>
</tr>
</tbody>
</table>

Match | Action
------|------
mac3  | port2
mac4  | port2
Gateway Cluster: Fast Recovery with Multi IPs

Multi IPs Mapping Table:

<table>
<thead>
<tr>
<th>Node</th>
<th>vIP</th>
<th>vMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node1</td>
<td>ip1</td>
<td>mac1</td>
</tr>
<tr>
<td></td>
<td>ip2</td>
<td>mac2</td>
</tr>
<tr>
<td>Node2</td>
<td>ip3</td>
<td>mac3</td>
</tr>
<tr>
<td></td>
<td>ip4</td>
<td>mac4</td>
</tr>
<tr>
<td>Node3</td>
<td>ip5</td>
<td>mac5</td>
</tr>
<tr>
<td></td>
<td>ip6</td>
<td>mac6</td>
</tr>
</tbody>
</table>

**Match** | **Action**
--- | ---
mac3 | port1
mac4 | port3

RARP

ToR Switch

vSwitch

VM VM ...

vSwitch

VM VM ...

vSwitch

VM VM ...

......
Management Layer: Cluster Mapping

- Intra-cluster load adjustment
  - Multi IPs planner
  - Rapid cluster scaling

- Inter-cluster load balancing
  - VPC-Cluster Mapping (ILP)
  - Solved by a rounding-based algorithm

More details in the paper
Experimental Setting

- **Testbed**
  - 45 KVM-based gateways on 3 bare metal servers
  - 20 bare metal servers as compute nodes

- **Comparison**
  - GWZone (conventional gateway model)
  - On-Demand Model
  - Preprogrammed Model
Evaluation

**Low-latency**
What’s the benefit and overhead of Zeta’s flow detection and rule offloading?

**Scalability**
Can Zeta provide load balancing for the gateway nodes?

**Robustness**
How well can Zeta handle failures?
Microbenchmark: Benefit of Doing Detection on Gateway

Faster rules offloading with flow detection on gateway
Microbenchmark: Overhead

Low overhead of flow detection and rule offloading on gateways

Packet Rate (Mpps)

- Forward Only
- Forward + Offload
- Forward + Detect
- Forward + Offload + Detect

No. of eBPF Map Entries
- 2k
- 10k
- 100k

1.86Mpps, 1.71Mpps
Evaluation: Load Balancing

- Video traffic
- Flow ranges from 2.4Mbps to 100Mbps

Both lower average load and better load balancing for gateways
Evaluation: Fast Recovery

• Fast Failure Recovery
  • Zeta achieves gateway failure recovery in 5.5ms through Multi IPs migration

• Intra-Cluster Load Adjustment
  • Multi IPs migration can quickly adjust the intra-cluster load

10.8X times faster failure recovery
## Conclusion

Zeta: a scalable and robust east-west communication framework in large-scale clouds

<table>
<thead>
<tr>
<th>Models</th>
<th>Robustness</th>
<th>Scalability</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprogrammed</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>On-Demand</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Gateway</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Zeta</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Thank you

https://github.com/futurewei-cloud/zeta/

For more questions:
zqy2019@mail.ustc.edu.cn