Simplify Distributed Rate-Limiting in Overlay Cloud Network with FDRL

Alibaba Cloud Senior Technical Expert, Stephen Xu
Agenda

- Alibaba cloud network infrastructure Introduction
- What’s problem we meet in Cloud overlay network?
- Key idea of FDRL
- Experiment in VPP
Global Alibaba Cloud Network Infrastructure

18 Region

110+ PoP

1500+ Edge
Alibaba Network Architecture
Apsara LuoShen, Alibaba Cloud SDN Architecture

Apsara LuoShen System

Control Plane
- VPC Controller
- CEN Controller
- CCN Controller
- NFV Controller

Data Plane
- Apsara vSwitch
- DCN GW
- Hybrid Cloud GW
- Internet GW
- SLB
- CCN

Management Plane
- Intelligent Maintenance System
- Intelligent Operation System
- Data Analysis System

Intelligent Maintenance System
Intelligent Operation System
Data Analysis System
LuoShen Evolving

- Classic
  - Network
    - First Gen
  - Connection
    - Classic

- VPC
  - Tenant
    - Separation
    - Second
  - VPC

- Cross Connect
  - Third
    - CEN
    - CCN
    - Branch
    - Store
    - IDC
    - US
    - VPC China
    - VPC Euro

- Networkless
  - The Next..
Comprehensive Networking Product Family

12 products for 5 scenarios

- **Self-Defined Cloud Networking Environment**: VPC
- **Internet Access**: SLB, NAT Gateway, EIP
- **Saving BGP Cost**: Shared Flow Package, Shared Bandwidth Package
- **Building Hybrid Cloud**: Express Connect, Cloud Hosting, VPN Gateway, Smart Access Gateway (CCN)
- **Global Connection**: Cloud Enterprise Network (CEN), Global Acceleration (GA)

From Data Center to Global Connection, for every enterprise networking scenarios
Problems?

VM Want to Split Internet Traffic Across Multiple underlay Paths to multiple IGWs

• Load balancing to different Path between VM and IGW cluster nodes.
• We focus on Overlay nodes, ignoring the underlay topology.

Underlay network is usually CLOS in DC deployment.

LB exactly do in **two Layers**:

• Underlay network, ECMP is done btw different physical paths.
• Overlay network, ECMP is done btw different IGWs within the cluster.

Key challenges are in overlay:

1. How to do the ECMP
2. How to do the rate-limiting
How to Split traffic to different Paths?

How to Split Traffic?

<table>
<thead>
<tr>
<th>Packet-Based</th>
<th>Flow-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Accurate</td>
<td>• Inaccurate</td>
</tr>
<tr>
<td>• Reorders TCP packets</td>
<td>• No packet reordering</td>
</tr>
<tr>
<td>• Easily tracks dynamic ratios</td>
<td>• Problem: Elephants Flow and Mice flows</td>
</tr>
</tbody>
</table>

Can we simply combine the best of the two approaches in overlay case?

- Load balancing to remove hot spots
- Problem: Elephant flow and Mice flow
- Rebalance traffic when unpredictable events occur (Outages, DoS, BGP reroutes, Flash Crowds, ...)

There is good idea of quickly bypassing the failure point through changing the overlay src_port (failure recovery)
How to make the rate-limiting?

Reroute and aggregate the same flow to one GW to do the centralized rate-limiting.

- Flow-based traffic Load balance;
- dynamically estimate the rate of each TCP flow
- automatically change the rate-limiting rates

(S1) Simple but not efficient

(S2) Too complex, may be not accurate
What is the FlowLet?

Flowlets exist because TCP is burst:
- TCP usually sends a window in one or a few bursts and waits for acks
- Slow-start
- Ack compression
- Window is much smaller than delay-BW product

Most flowlets have inter-arrivals less than an RTT
→ most flowlets are sub-windows

Two Cisco papers:
Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching
CONGA: Distributed Congestion-Aware Load Balancing for Datacenters
FDRL (Flowlet based distributed rate-limiting)

- Select the static Time-Diff as 300us or 100us
- In the VM side, virtual switch will do the FlowLet splitting
- Controller to dynamically change the rate-limiting ratio according to the cluster member changing

\[ \delta > |D2 - D1| \]

\[ \text{Delay} = D1 \]
\[ \text{Delay} = D2 \]
FDRL leverage the simple FlowLet mechanism using the overlay UDP src port as the entropy.

- Simply implement
- Self-adaption
- Great performance improvement
- Great BW coefficient of utilization
How to implement the FDRL in VPP

Base on the flowtable plugin

- Support Dynamical session for TCP flow, with session aging/timeout
- TCP stateful session, TCP state update.
- Record the timestamp for last pkt receiving based on flow
- Make the judgement for the action: when we need to change the path
- How to change the path: just change the overlay udp src port.

Implementing Flowlet in VPP is Simple

- Record the flow and timestamp for the time of last receive pkt
- If (Now - Last_Seen) > δ, flow can change path
- Change the overlay encap filed of src_port
- Reassign path proportionally to the desired split ratios

<table>
<thead>
<tr>
<th>SRCIp</th>
<th>DSTip</th>
<th>SRCPort</th>
<th>DSTPort</th>
<th>hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>9920.2659</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
$ vpp show flowtable
Number of flows cache allocated:256
active_flow: 0
run show cmd time(s):6829297
...
$ vpp show flowtable | grep 9920.2659
Number of flows cache allocated:256
active_flow: 0
run show cmd time(s):6829297
```

```
<table>
<thead>
<tr>
<th>Plugin</th>
<th>Source</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpp.so</td>
<td></td>
<td>18.07.7-glibsslf-dirty</td>
<td>Rocket Memory Inter</td>
</tr>
<tr>
<td>vpp.so</td>
<td></td>
<td>18.07.7-glibsslf-dirty</td>
<td>Intel Adaptive Virt</td>
</tr>
</tbody>
</table>
```

```
$ ip -s src 10.0.1.26 dst 10.0.1.32 proto 6 sport 4444 dport 80 src_port 9920.2659 dst_port 8080
$ vpp show flowtable
```

```
$ vpp show flowtable
```

```
$ vpp send -m echo ip 10.0.1.26 10.0.1.32 proto 6 src_port 9920.2659 dst_port 80 dst_ip 192.168.1.10
```
How to implement the FDRL in VPP

Base on the flowtable plugin

- Support Dynamical session for TCP flow, with session aging/timeout
- TCP stateful session, TCP state update.
- Record the timestamp for last pkt receiving based on flow
- Make the judgement for the action: when we need to change the path
- How to change the path: just change the overlay udp src port.
What is the next Step?

Call for community to join this direction, and make more improvement:

• Dynamically change the time diff for different Elephants/Mice flow

• Asymmetry rate-limiting scenario

• Flowlet is still working or not after BBQ?

• Use the same logic to quickly bypass the failure path through changing the overlay src-port

• VPP implementation, and integration to support more FDRL features
Our Vision

Simply The Network

[QR Code] Blog
Scan to Learn More

[QR Code] DingDing User Group
Scan to Learn More
Thanks