OpenStack on L3 Clos-NW First-Hop Redundancy by Open vSwitch & BGP in Yahoo! JAPAN network.

Cloud Network Team

Yusuke Tatsumi (立見 祐介)

LINEヤフー

Please find the English slide after Japanese one.

何の話？

HypervisorのOVS/BGPでのNW冗長化設計の話

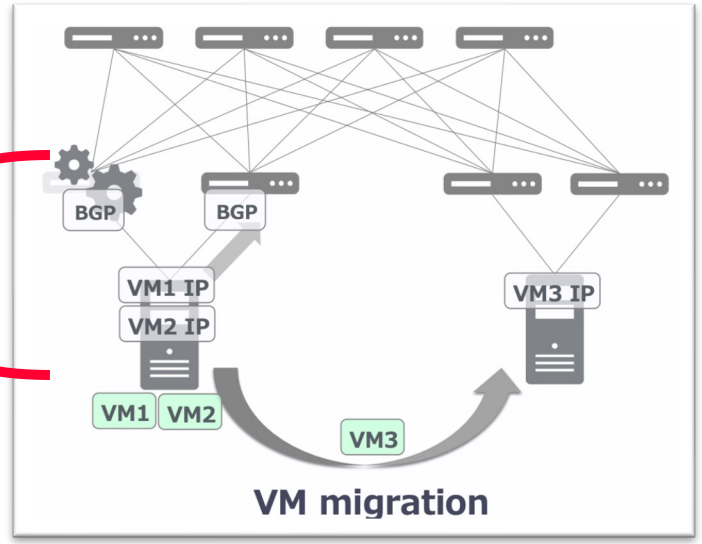
これまでのJANOGセッションとの比較

今回の話

Yahoo! JAPANの仮想環境にて、
L3 Clos-NW上で、
HV first hopをOVS/BGPで冗長化
この設計で約5年運用中。

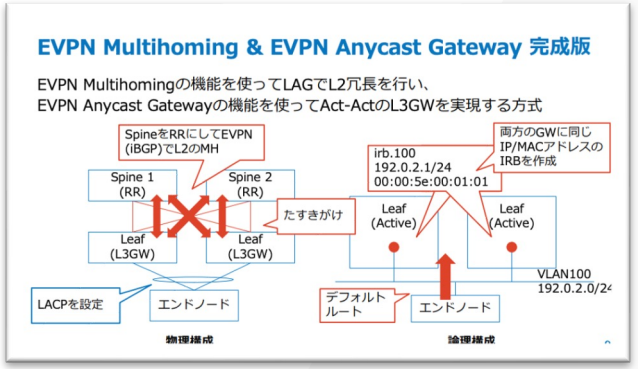
※ NOT LINE, NOT LYの話です。

@JANOG46 ヤフーの IP Closネットワークの歴史と運用
（ALL L3 BGP Clos Networkの話）



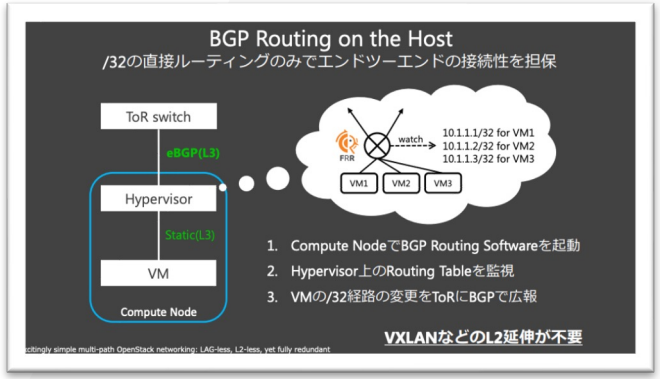
https://www.janog.gr.jp/meeting/janog46/yahoo/

@JANOG47 みんなFirst-Hop Redundancyどうしてるよ？
（EVPN MH&AGでの設計）



https://www.janog.gr.jp/meeting/janog47/fhrp/

@JANOG43 LINEのネットワークをゼロから再設計した話
（All L3 Clos NW with Linux networking）



https://www.janog.gr.jp/meeting/janog43/program/line/

全体設計

1. ゲスト側: Open vSwitchを利用（OVS, OVS/DPDK, OVS-TC）

VMのIPを冗長広報
- In-house agentが広報

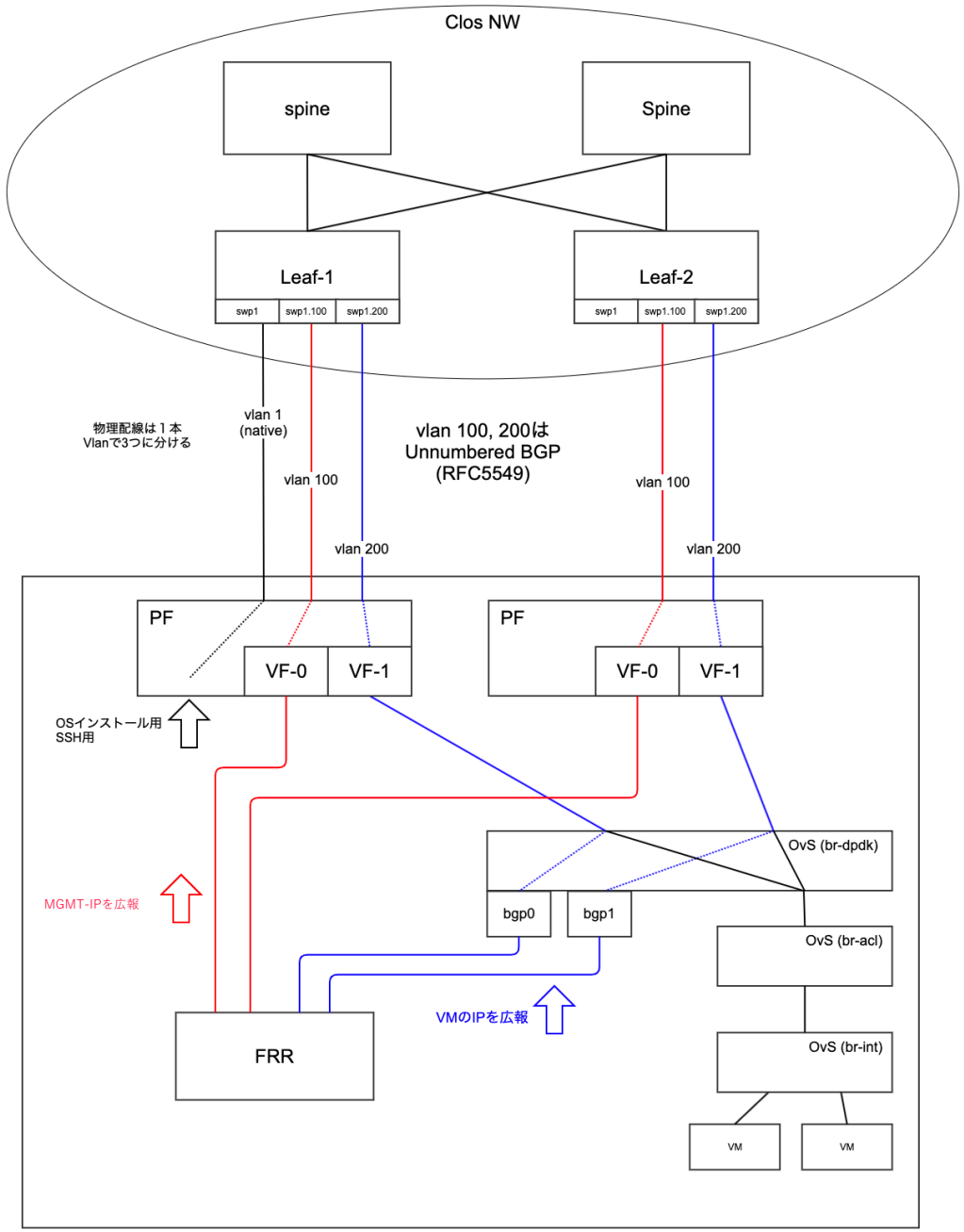
-> OVSのgroupとbucketで冗長

2. ホスト側: Linux Networkingを利用

通常利用IP (MGMT-IP)を冗長広報
- OS起動時にchefで広報開始
- 各コンポーネント・監視・ssh等全ての通信で利用

-> ip ruleとrouting table分割で冗長

OSインストール時のIPは片系のみ
- I/FへIPアドレスを”普通に”付与
- 初期setup時やOS焼き直し時のみ利用
- 通常時は使用しない



1. ゲスト側

group/bucketで冗長化

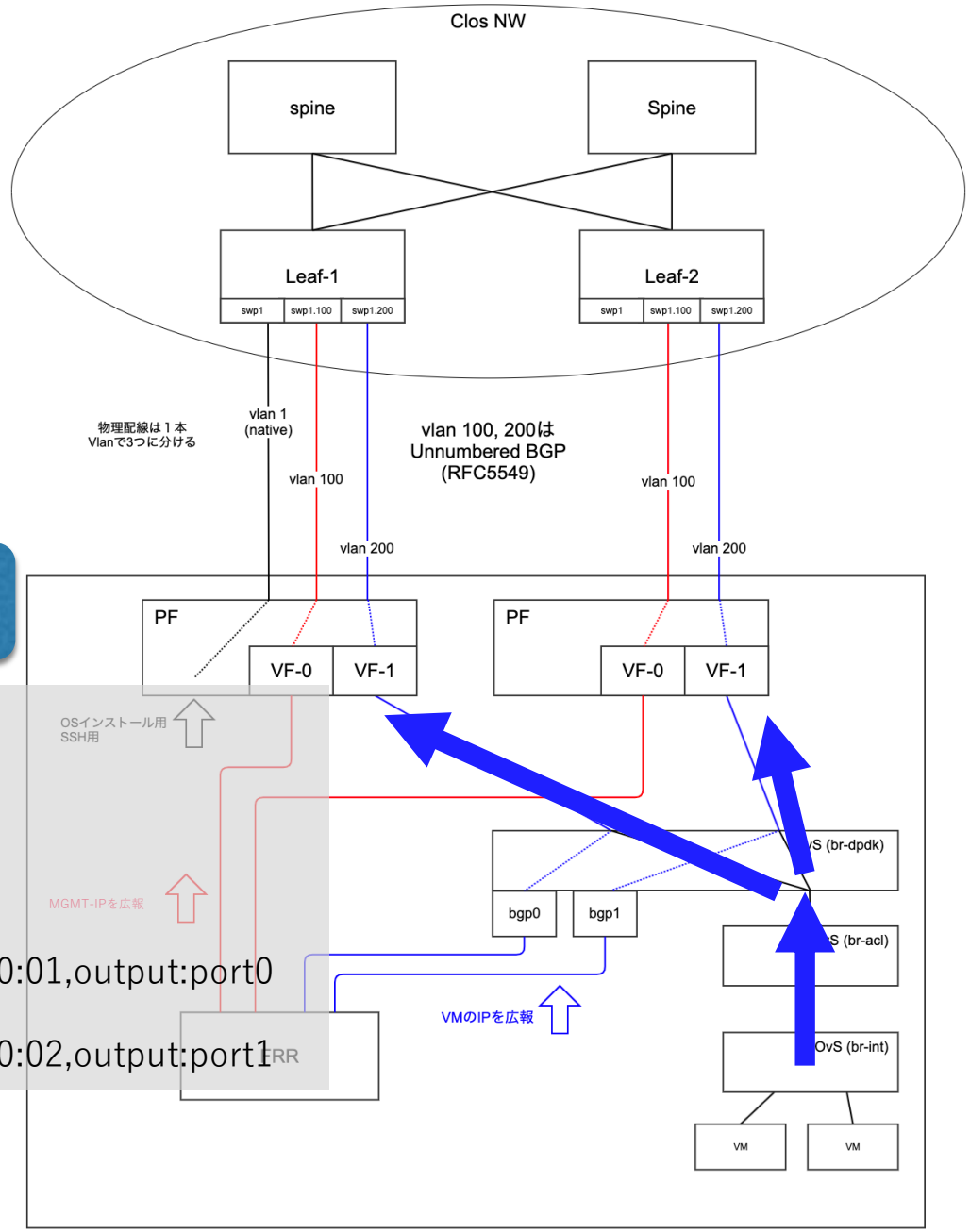
```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
```

外に出る場合は
group:1を参照

group_id=1: bucketの候補
(port0とport1)で通信冗長

```
$ sudo ovs-ofctl -O OpenFlow11 dump-groups br-dpdk

group_id=1,type=select,selection_method=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```

group_id=2, 3: bucketは片系の出口のみ定義



OVSトラフィックの寄せは？障害時は？

ゲスト側

メンテ等でのトラフィック寄せ

1. ToR SW側にてBGP graceful-shutdown発行。
2. ホスト側daemonがこれを検知し、flow ruleを一時的に追加。

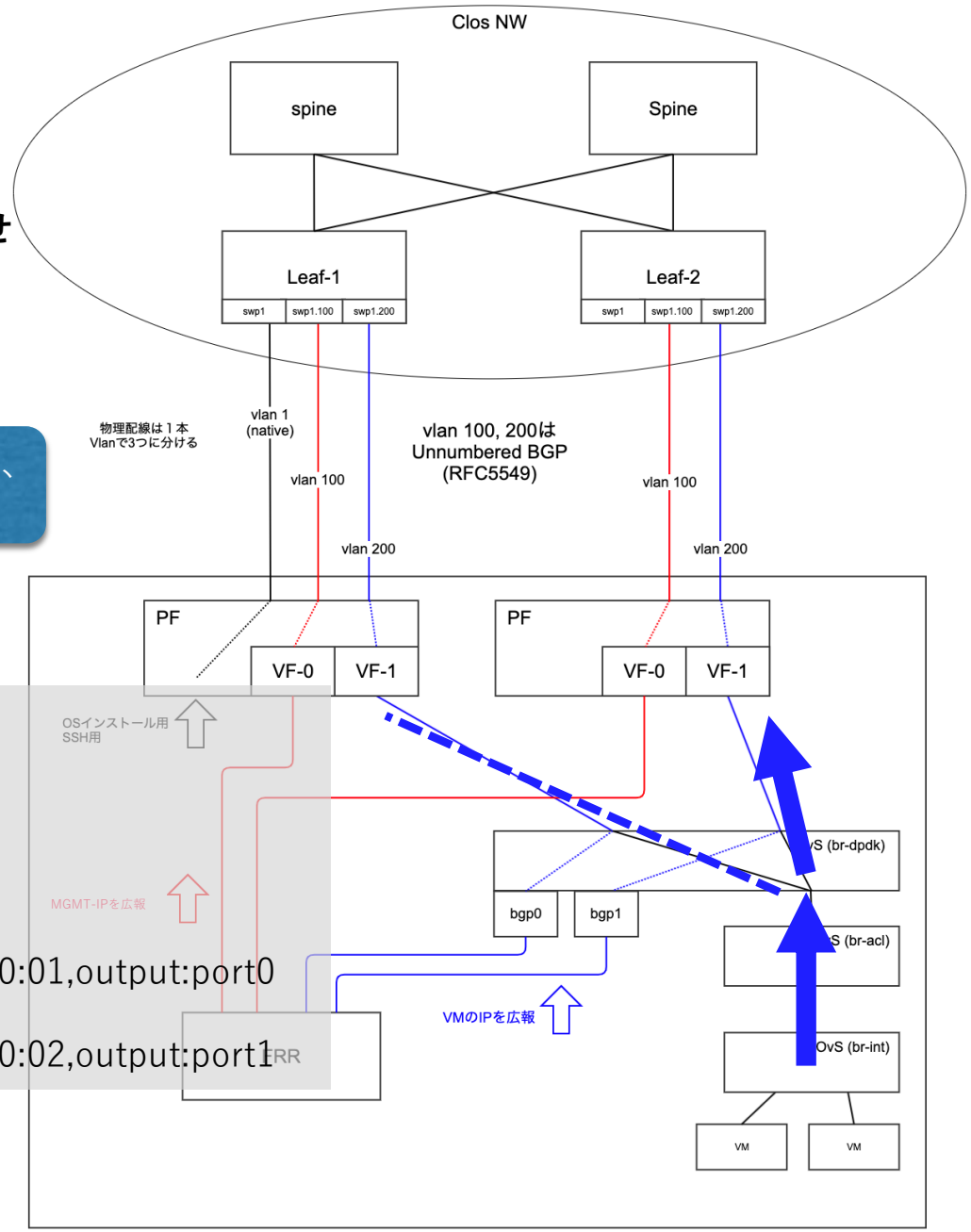
```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
priority=750,in_port="from-br-acl" actions=group:2
```

高優先度ruleを自動追加し、迂回用groupを利用

```
$ sudo ovs-ofctl -O OpenFlow11 dump-groups br-dpdk

group_id=1,type=select,selection_method=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```

group_id=2が参照される

通信影響無し！



ゲスト側

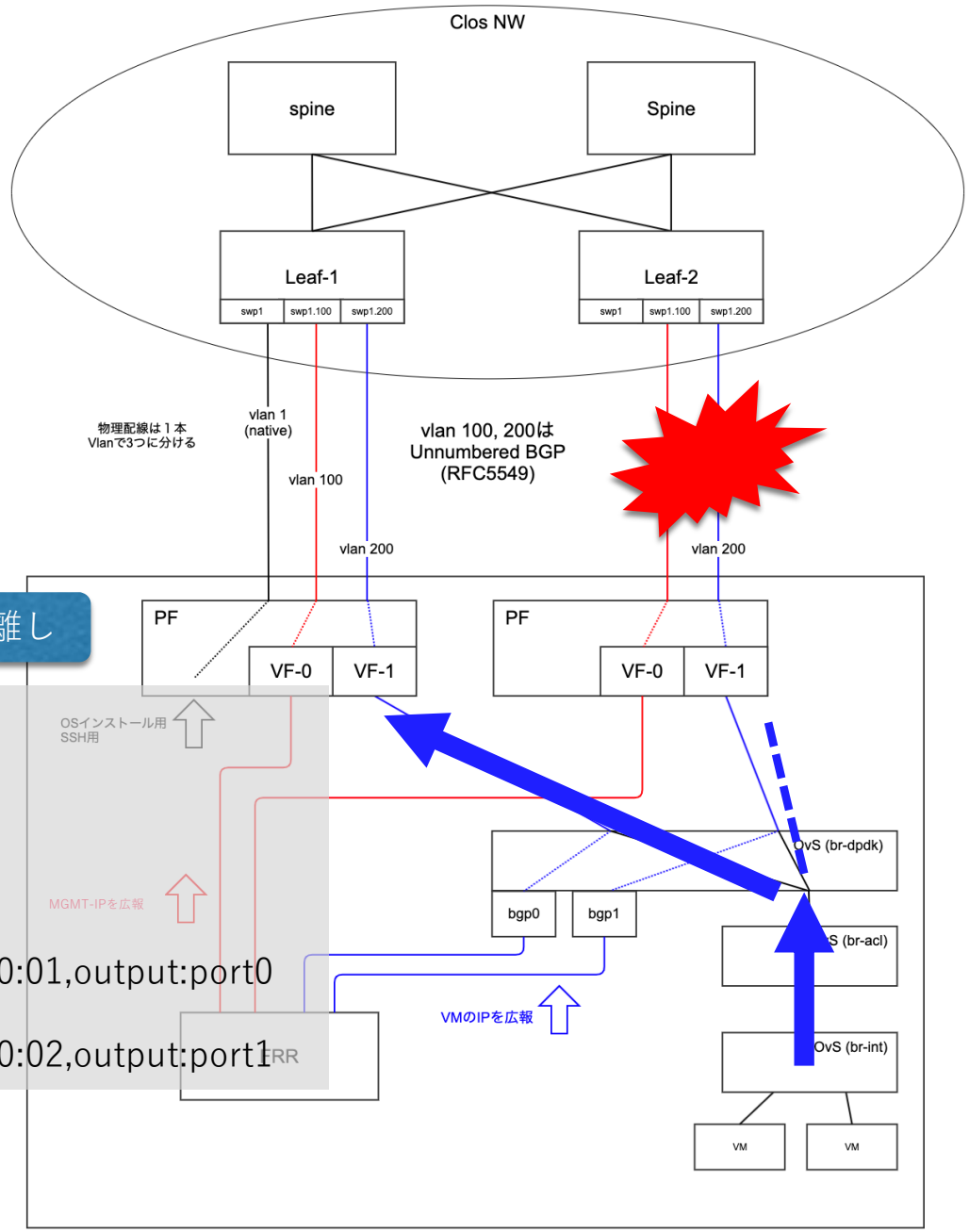
I/F down時。

1. ホスト側で対向link down検知。
2. OvS watch_port機能で通信縮退。

```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
```

watch_port機能でlink-downしたI/F (bucket)を自動切り離し

```
$ sudo ovs-ofctl -O OpenFlow11 dum          br-dpdk

 group_id=1,type=select,selectio     ethod=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```



0.5秒程度のpingロス

ホスト側

ip ruleとrouting table分割で冗長化

```
$ ip rule
0:      from all lookup local
20:     from 192.168.1.2 lookup main
30:     from all lookup 200
32766:  from all lookup main
32767:  from all lookup default
```

OS付与IPアドレスから出る場合は mainテーブルを参照

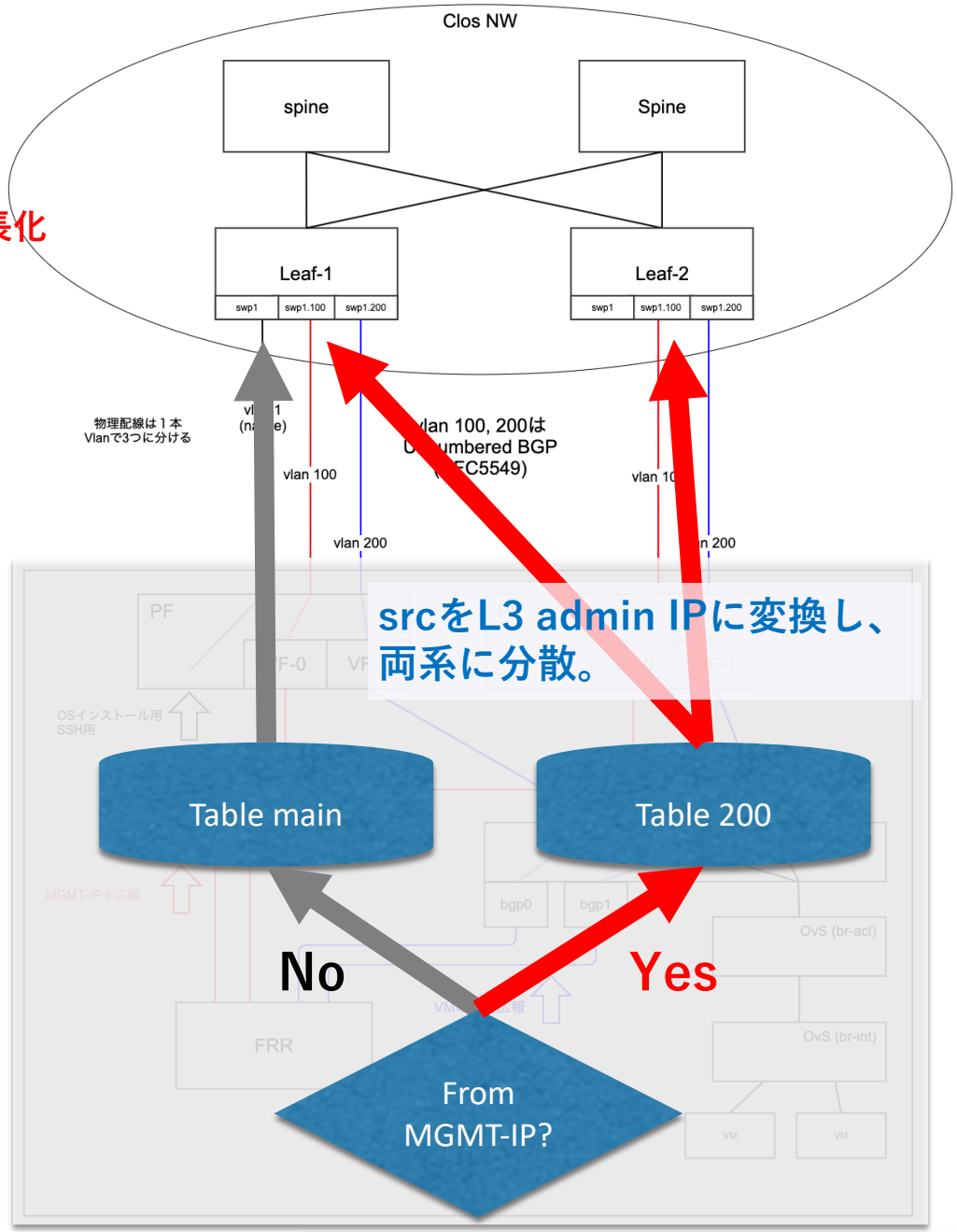
その他すべての通信はtable 200を参照

```
$ ip route show table main
default via 192.168.1.1 dev eth0
```

片系Leaf/NICのみ利用

```
$ ip route show table 200
default src 172.16.1.2
        nexthop via 169.254.0.1 dev vf_bgp0 weight 1
        nexthop via 169.254.0.1 dev vf_bgp1 weight 1
```

Src IPをMGMT-IPに変換し、両系Leaf/NICを利用



Lessons Learned (1)

2019年から5年以上運用中。
HV 20k台, VM 200k台規模で稼働中。

FRR watchdog問題

- systemctl realod frrとdaemon-reloadを同時実行すると高確率でsystemdが死ぬ
- systemdのバグ起因。FRRのwatchdogsec=0(無効化)で一時対処。
- Issue: https://github.com/systemd/systemd/issues/15356 & https://github.com/systemd/systemd/pull/15370
- Fix: https://git.centos.org/rpms/systemd/c/e8f8de55659f4883530960b5b591eda842b35746

FRRバージョン差異 (FRR7系、8系)によるconfig/設定方法の差異

VM IP広報重複問題

- サーバ側で確認できるが、NW側（Clos-NW全体）でも正常性確認が必要

事前・事後の継続的な動作確認が重要。

Lessons Learned (2)

BGP TAP I/F遅延問題

- ACL更新 -> TAP I/Fだけ遅延 -> BGP timer切れ -> VM疎通断
- BGP Hold-down timer延長, TAP PMD化 (DPDK化)

vswitchdだけ落ちる問題

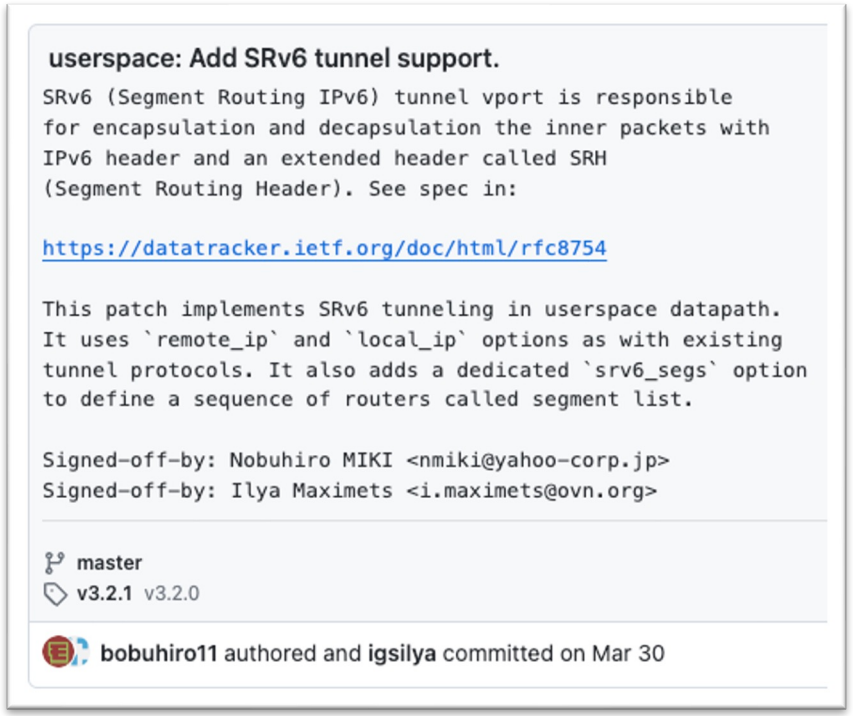
- CentOSはovsdbとvswtichdを1unitで管理
  -> 片方だけ死んでも再起動させてくれない
- OvS-TC HW offloadで半死
  - サービス通信断（first packetをチェックするvswitchdが不在）
  - ヘルスチェックは無事 (継続通信は無事)

NICドライバのバグ・・・

設計PoC時に性能限界を試す事が重要。

OSSを使うだけではなく、upstreamもやってます！

- OVS SRv6 tunnel implementation
  - OVS v3.2.0から利用可能！
  - OVSCONにて登壇



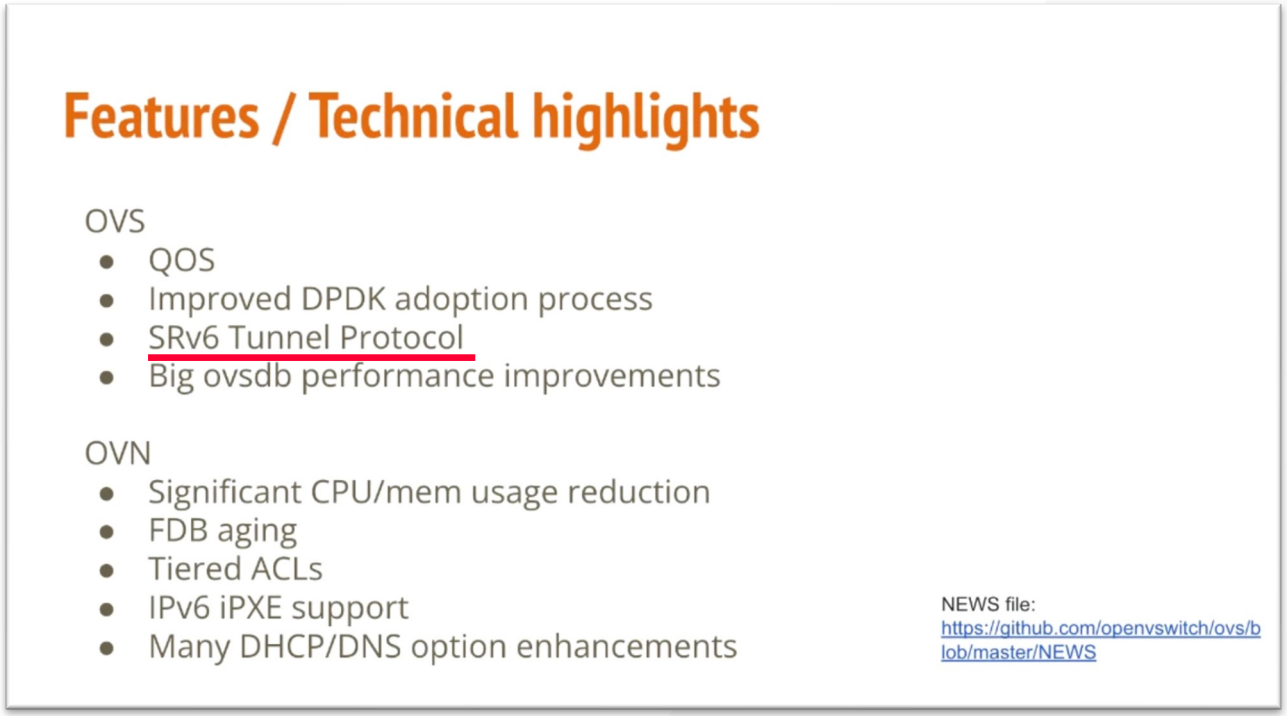
userspace: Add SRv6 tunnel support.

```
SRv6 (Segment Routing IPv6) tunnel vport is responsible
for encapsulation and decapsulation the inner packets with
IPv6 header and an extended header called SRH
(Segment Routing Header). See spec in:

https://datatracker.ietf.org/doc/html/rfc8754

This patch implements SRv6 tunneling in userspace datapath.
It uses `remote_ip` and `local_ip` options as with existing
tunnel protocols. It also adds a dedicated `srv6_segs` option
to define a sequence of routers called segment list.

Signed-off-by: Nobuhiro MIKI <nmiki@yahoo-corp.jp>
Signed-off-by: Ilya Maximets <i.maximets@ovn.org>
```

master
v3.2.1 v3.2.0

bobuhiro11 authored and **igsilya** committed on Mar 30

https://github.com/openvswitch/ovs/commit/03fc1ad78521544c7269355ec72fec8c2373b96d



Features / Technical highlights

OVS
- QOS
- Improved DPDK adoption process
- SRv6 Tunnel Protocol
- Big ovsdb performance improvements

OVN
- Significant CPU/mem usage reduction
- FDB aging
- Tiered ACLs
- IPv6 iPXE support
- Many DHCP/DNS option enhancements

NEWS file:
https://github.com/openvswitch/ovs/blob/master/NEWS

https://www.openvswitch.org/support/ovscon2023/

ご清聴ありがとうございました！

続きはLY展示ブースにて会話させてください！！！

LINEヤフー

English slide.

What are you talking about?

Hypervisor’s NW redundancy design w/ OVS/BGP.

Comparison with previous JANOG sessions.

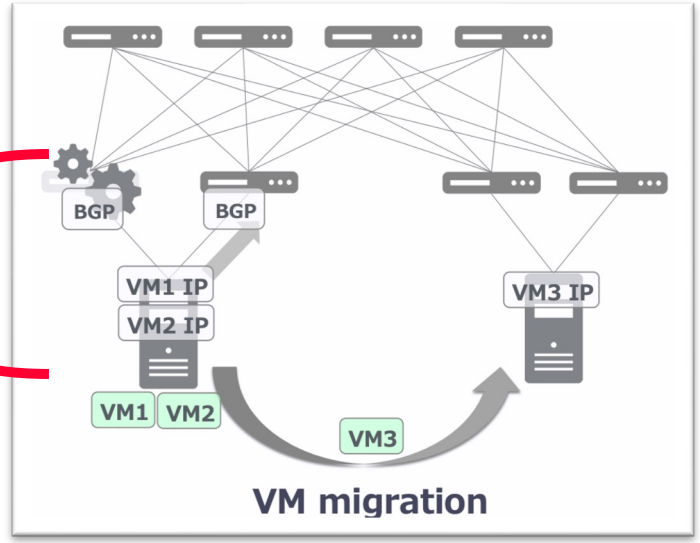
Today‘s topic:

For Yahoo! JAPAN IaaS on top of L3 Clos-NW, HV first-hop redundancy using OVS/BGP.

5 years in production.

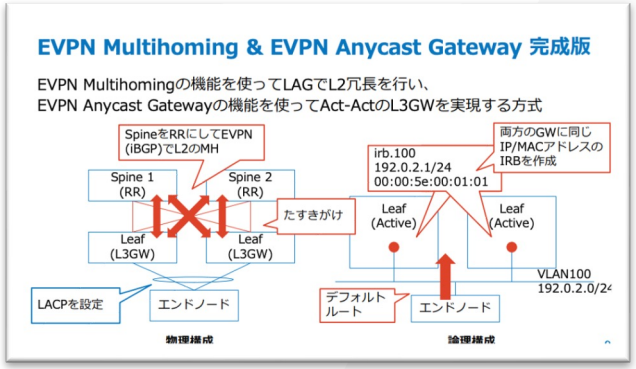
※ NOT LINE, NOT LY design topic.

@JANOG46 History and Operation of Yahoo's IP Clos Network
（ALL L3 BGP Clos Network）



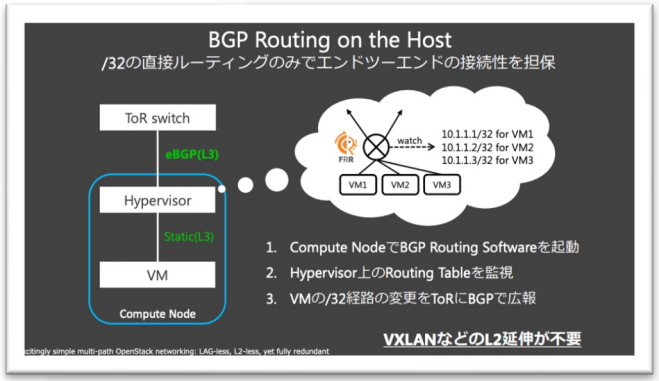
https://www.janog.gr.jp/meeting/janog46/yahoo/

@JANOG47 How is everyone doing with First-Hop Redundancy?
（EVPN MH&AGでの設計）



https://www.janog.gr.jp/meeting/janog47/fhrp/

@JANOG43 The story of the redesign of LINE's network from scratch.
（All L3 Clos NW with Linux networking）



https://www.janog.gr.jp/meeting/janog43/program/line/

Overall Design

1. Guest side(VM): Open vSwitch（OVS, OVS/DPDK, OVS-TC）

Redundant advertisement of VM's IP
- Adv. by In-house agent

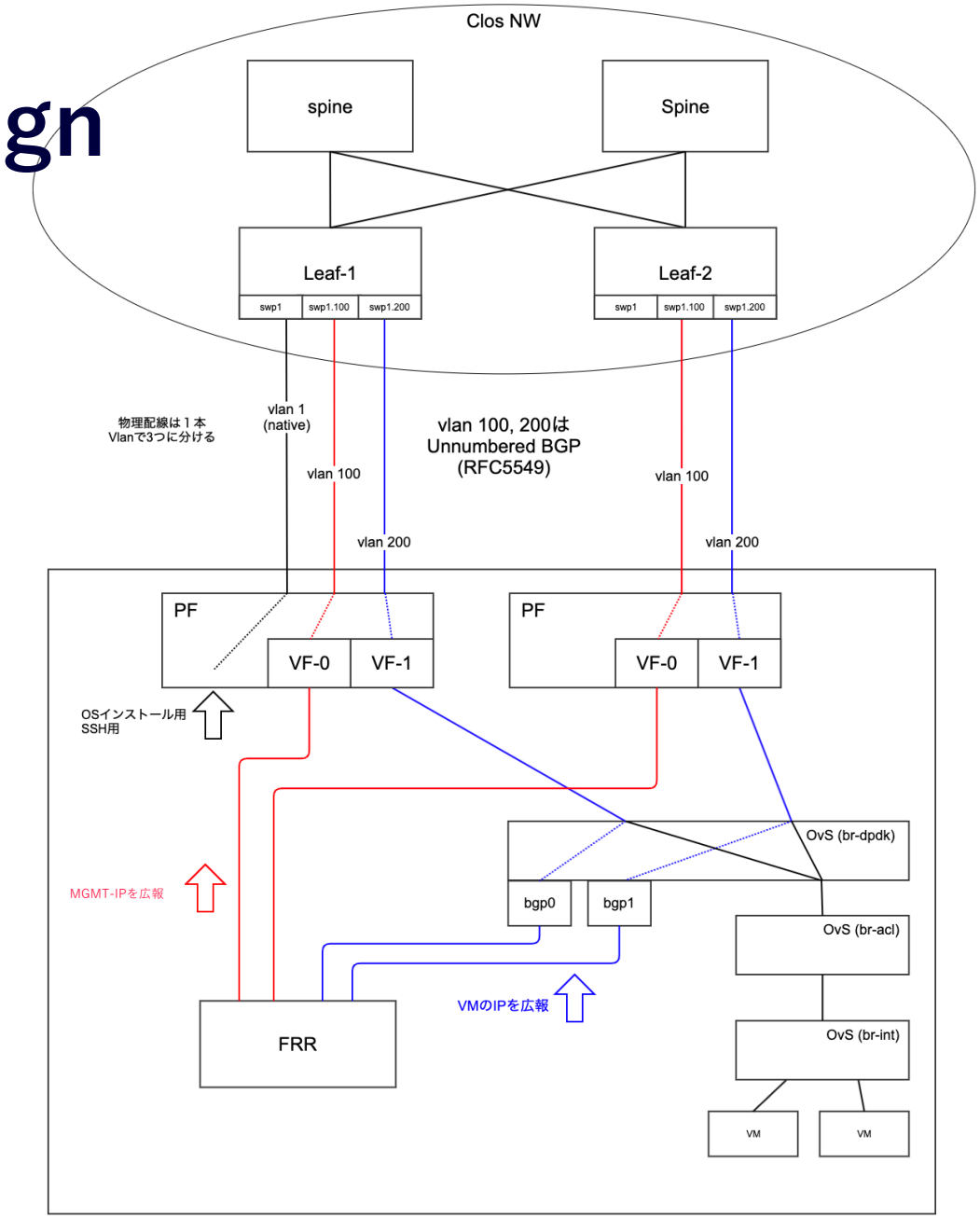
-> Redundant by OVS group & bucket

2. Host side: Linux Networking

Redundant advertisement for management IP (MGMT-IP)
- Adv. is started by chef at OS startup
- Used for all component, monitoring, ssh, etc.

->Redundancy by ip rule & routing tables

Non-redundancy IP is used for only OS installation.
- IP address is assigned to I/F normally
- It's used only during initial setup and OS rebuild
- Not used during normal operation



1. Guest side

Redundancy by group/bucket.

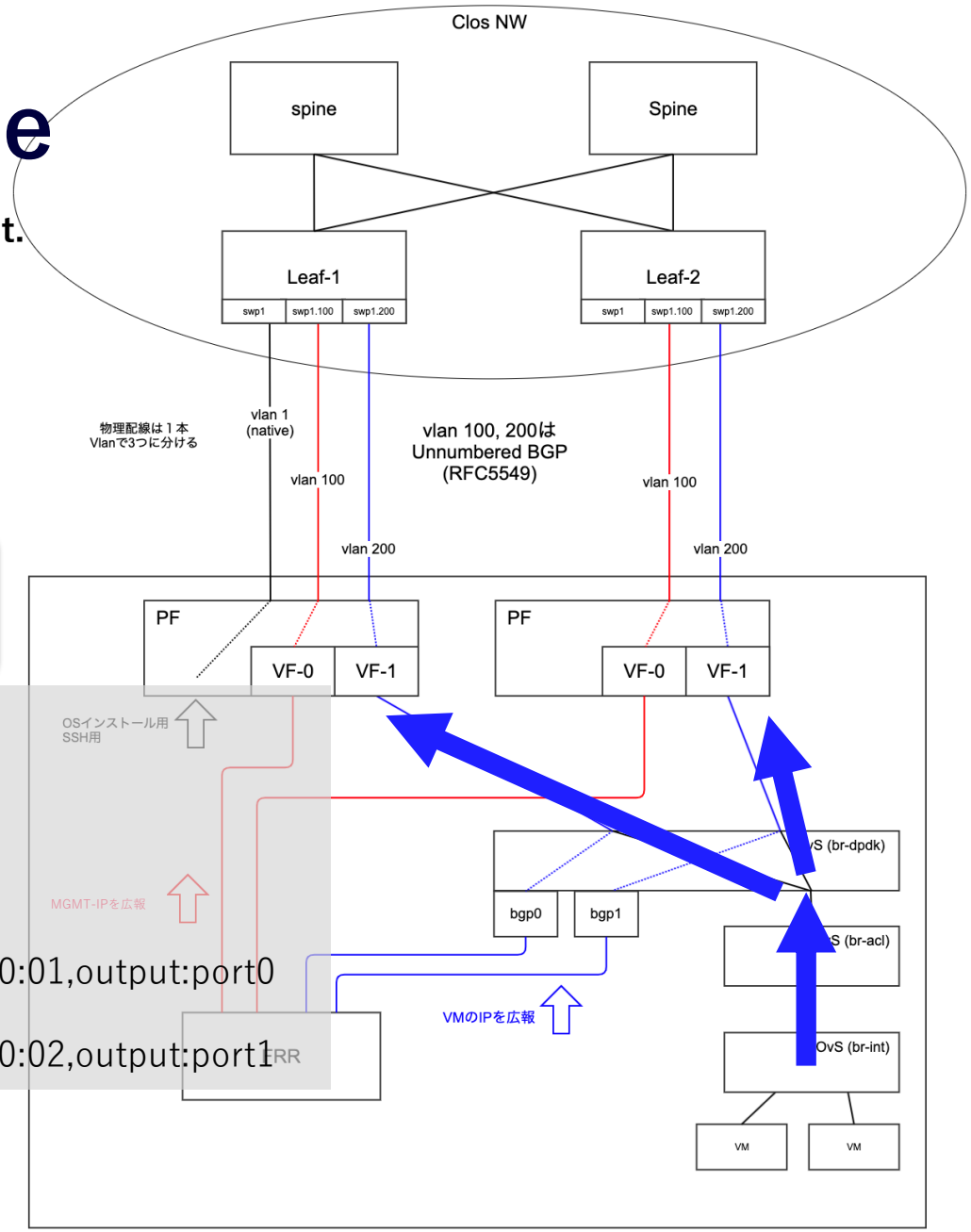
```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
```

Egress case:
Lookup group:1

group_id=1: Bucket candidates (port0とport1) for redundancy.

```
$ sudo ovs-ofctl -O OpenFlow11 dump-groups br-dpdk

group_id=1,type=select,selection_method=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```

group_id=2, 3: Bucket is defined only for one exit.



What about scheduled/unscheduled maintenance for OVS?

Gues side

Scheduled maintenance case.

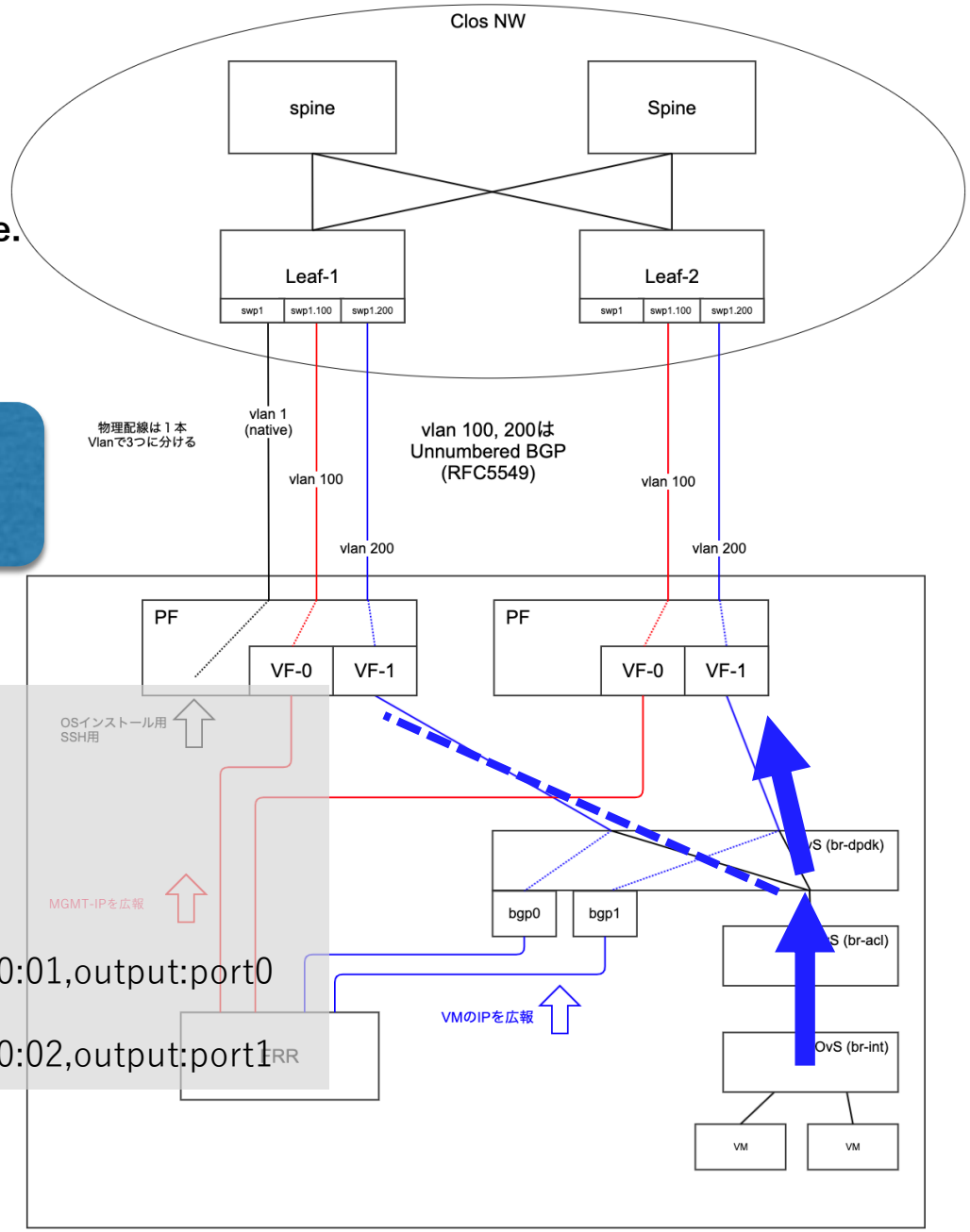
1. BGP graceful-shutdown is issued on the Tor SW side.
2. Host-side daemon detects this and adds a flow rule temporarily.

```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
priority=750,in_port="from-br-acl" actions=group:2
```

Automatically add higher priority rules and use groups for bypassing.

```
$ sudo ovs-ofctl -O OpenFlow11 dump-groups br-dpdk

group_id=1,type=select,selection_method=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```

Lookup group_id=2



No packet loss!

Guest side

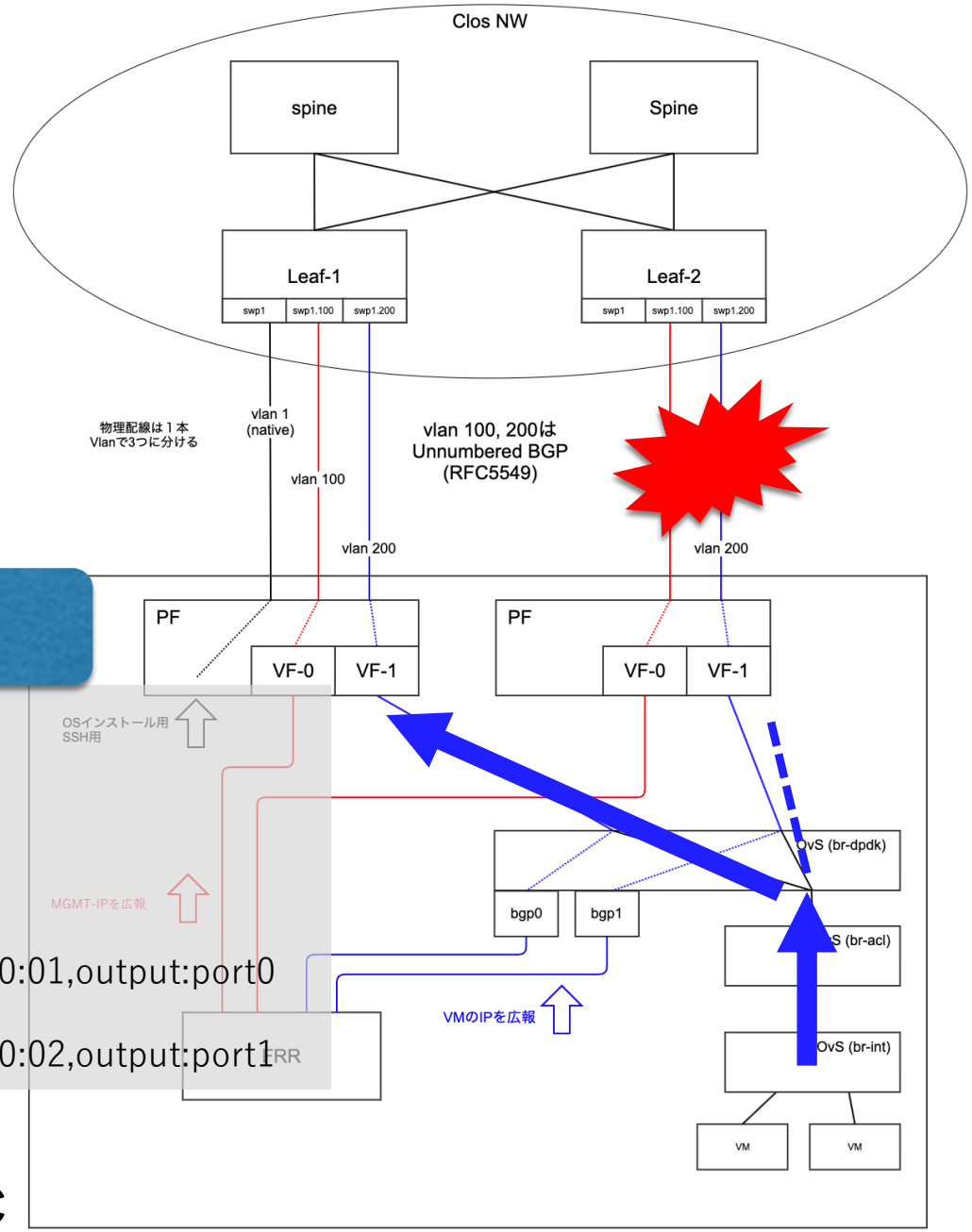
I/F down case.

1. Host side detects a link down.
2. Detour by OvS watch_port function.

```
$ sudo ovs-ofctl dump-flows br-dpdk
priority=700,in_port="from-br-acl" actions=group:1
```

Automatic disconnection of failure I/F (bucket) using watch_port function

```
$ sudo ovs-ofctl -O OpenFlow11 dump-group-stats br-dpdk

group_id=1,type=select,selection_method=hash,fields(ip_dst,ip_src), \
bucket=watch_port=port0,set_field=80:a2:35:00:00:01 ->eth_dst,output:port0, \
bucket=watch_port=port1,set_field=80:a2:35:00:00:02 ->eth_dst,output:port1

group_id=2,type=select,bucket=watch_port:port0,actions=mod_dl_dst:80:a2:35:00:00:01,output:port0

group_id=3,type=select,bucket=watch_port:port1,actions=mod_dl_dst:80:a2:35:00:00:02,output:port1
```



Ping loss of about 0.5 sec

Host side

Redundancy by ip rule and routing tables.

```
$ ip rule
0:      from all lookup local
20:     from 192.168.1.2 lookup main
30:     from all lookup 200
32766:  from all lookup main
32767:  from all lookup default
```

Lookup main table if coming from OS's IP address.

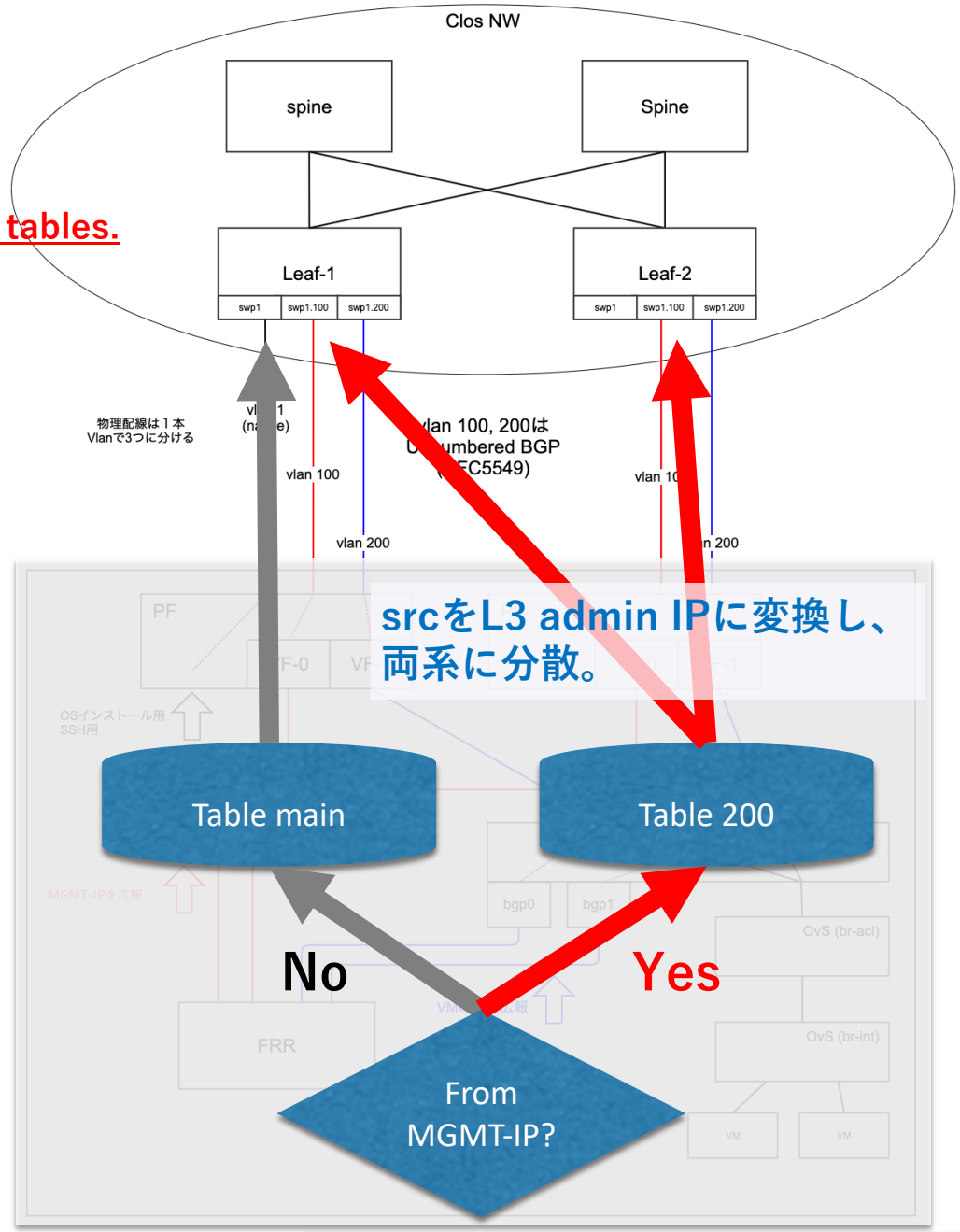
For all other communications, lookup table 200.

```
$ ip route show table main
default via 192.168.1.1 dev eth0
```

Only one Leaf/NIC is used.

```
$ ip route show table 200
default src 172.16.1.2
        nexthop via 169.254.0.1 dev vf_bgp0 weight 1
        nexthop via 169.254.0.1 dev vf_bgp1 weight 1
```

Convert Src-IP to MGMT-IP and use both leafs/NICs.



Lessons Learned (1)

In production for over 5 years since 2019.
Operating on a scale of 20k HVs, 200k VMs.

FRR watchdog issue

- High probability of systemd dying when systemctl realod frr and daemon-reload are executed at the same time.
- Due to systemd bug. Workaround: FRR watchdogsec=0 (disabled).
  - Issue: https://github.com/systemd/systemd/issues/15356 & https://github.com/systemd/systemd/pull/15370
  - Fix: https://git.centos.org/rpms/systemd/c/e8f8de55659f4883530960b5b591eda842b35746

FRR config/configuration incompatible issue for FRR7, FRR8.

VM's IP advertisement duplication problem.

- Should be checked not only on the server side but also on the Clos-NW side.

Continuous pre- & post-operational checks are important.

Lessons Learned (2)

BGP TAP I/F delay problem

- ACL update -> TAP I/F delay -> BGP timer expired -> VM disconnection…
- Solution: BGP Hold-down timer extension. Using TAP PMD for BGP I/F.

“Only vswitchd is down” gray failure.

- Ovsdb and vswtichd are managed in one system unit on CentOS
  - -> if one of them dies, it doesn’t restart automatically…
- OvS-TC HW offload causes gray failure.
  - Service traffic breakdown (vswitchd that checks first packet is absent)
  - Health-check itself is alive (Continuous communication is alive)

Bug in NIC driver…

It’s important to test the performance limit during design PoC.

Not only use OSS, but also upstream activity.

- OVS SRv6 tunnel implementation
  - Available since OVS v3.2.0 !

userspace: Add SRv6 tunnel support.

SRv6 (Segment Routing IPv6) tunnel vport is responsible for encapsulation and decapsulation the inner packets with IPv6 header and an extended header called SRH (Segment Routing Header). See spec in:

https://datatracker.ietf.org/doc/html/rfc8754

This patch implements SRv6 tunneling in userspace datapath. It uses `remote_ip` and `local_ip` options as with existing tunnel protocols. It also adds a dedicated `srv6_segs` option to define a sequence of routers called segment list.

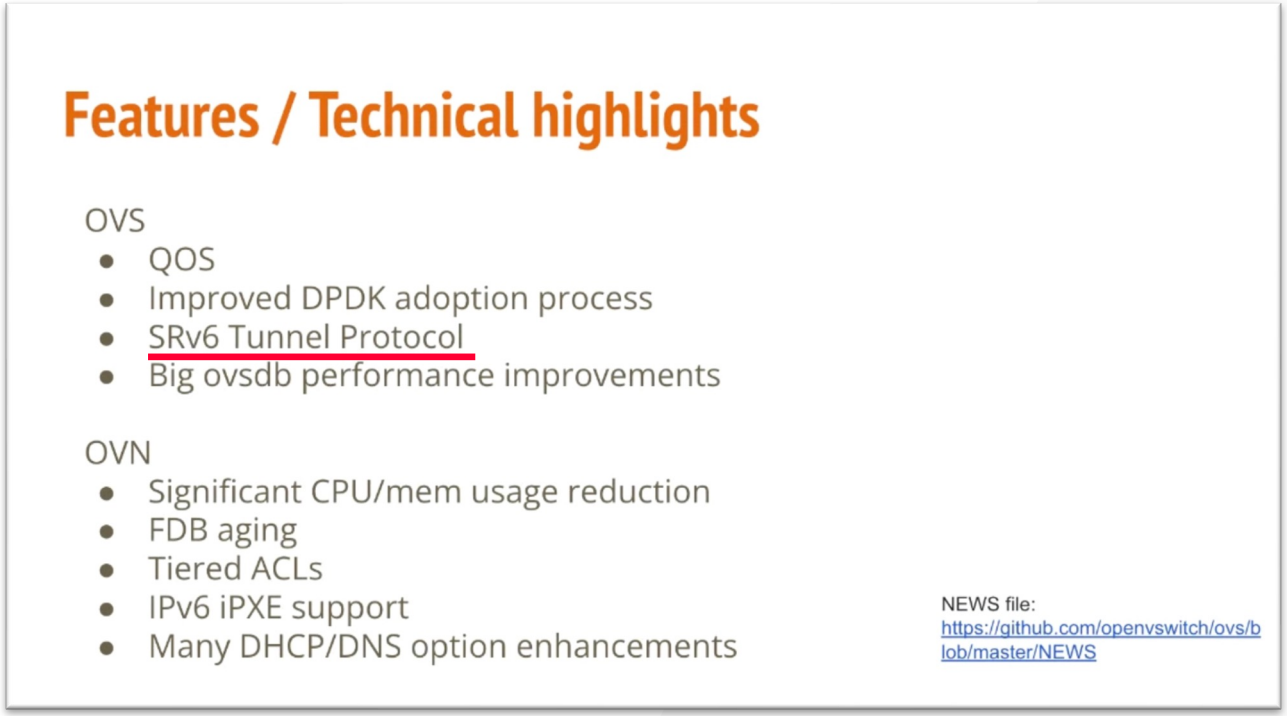
Signed-off-by: Nobuhiro MIKI <nmiki@yahoo-corp.jp>
Signed-off-by: Ilya Maximets <i.maximets@ovn.org>

master
v3.2.1 v3.2.0

bobuhiro11 authored and **igsilya** committed on Mar 30

https://github.com/openvswitch/ovs/commit/03fc1ad78521544c7269355ec72fec8c2373b96d

- Speaker at OVSCON 2023.



Features / Technical highlights

OVS
- QOS
- Improved DPDK adoption process
- SRv6 Tunnel Protocol
- Big ovsdb performance improvements

OVN
- Significant CPU/mem usage reduction
- FDB aging
- Tiered ACLs
- IPv6 iPXE support
- Many DHCP/DNS option enhancements

NEWS file:
https://github.com/openvswitch/ovs/blob/master/NEWS

https://www.openvswitch.org/support/ovscon2023/

Thank you for your attention!

Please continue the discussion at the LY exhibit booth!

LINEヤフー