JANOG44

OSSなWhitebox用NOSのSONiCが商用で使われている理由を考える

次世代技術本部　桑田　斉

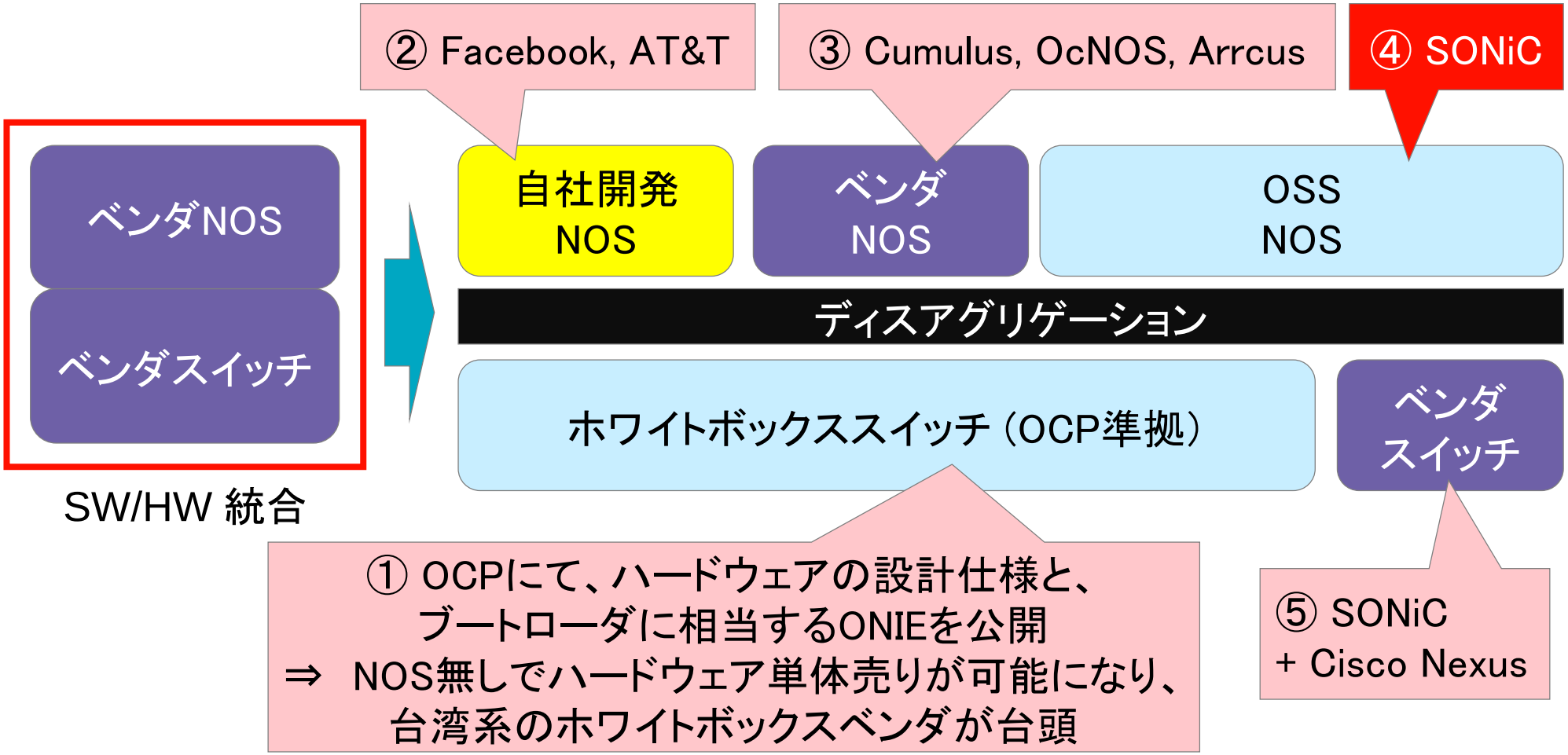
hitoshi.kuwata.gt@apresiasystems.co.jp

自己紹介

- 桑田　斉　（くわた　ひとし）
- APRESIA Systems 株式会社　次世代技術本部　技術開発部

- 2003年 日立電線株式会社※入社　　※ APRESIA Systemsのルーツ
  - 以来、APRESIAスイッチシリーズの製品ソフトウェア開発に従事
  - 弊社が昨年Edgecore NetworksのPremium Distributorになったことから、ホワイトボックススイッチ、特にBarefoot　Tofinoチップ搭載のWedgeBFによるP4プログラムが最近のメイン活動
  - 過去のJANOGでの発表
    - JANOG43 SRv6でサービスチェイニングをやってみた
      - https://www.janog.gr.jp/meeting/janog43/program/sc

ホワイトボックススイッチとNOSの関係



※OCP (Open Compute Project)
※NOS (Networking Operating System)
LinuxのOS部分とネットワーク機能のコントロールプレーンをまとめたソフトウェア

ハード・NOS組合せ比較

	ベンダNOS ベンダスイッチ	自社開発NOS ホワイトボックス	ベンダNOS ホワイトボックス	OSS NOS ホワイトボックス
NOS例	ベンダスイッチ 独自NOS	FBOSS dNOS	Cumulus, OcNOS Arrcus	SONiC
適用事例		Facebook AT&T	IIJ, LINE さくらインターネット Yahoo Japan	Microsoft Alibaba LinkedIn
NOSの カスタマイズ	不可	可	不可	可
導入コスト	中	大 NOS自社開発	小～中 NOSライセンス料	小 NOSが無償
ベンダ サポート	NOS/HW両方 サポート有り	無し 自己責任	NOS部分は サポート有り	無し 自己責任

① カスタマイズ性とコスト面ではベター （↑）
② 自己責任のため、運用側の負担・技術力が問われる （↓）
⇒ 本日はこれらの点について議論させていただきます

プレゼン内容と議論したいこと

APRESIA

◆プレゼン内容

- ◇ SONiCの紹介
- ◇ デモ① KVM環境でSONiC仮想マシンによるIP CLOSファブリック + Telemetry
- ◇ デモ② 実機SONiC (AS7712とWedgeBF32)によるIP CLOSファブリック

◆議論いただきたいこと

- ◇ White boxスイッチを運用することの問題・課題は何でしょうか？
- ◇ NOSの観点でも、ハードウエアの観点でも、ご意見お願いします！
- ◇ 逆に疑問点ありましたら、ぶつけていただいてもOKです！

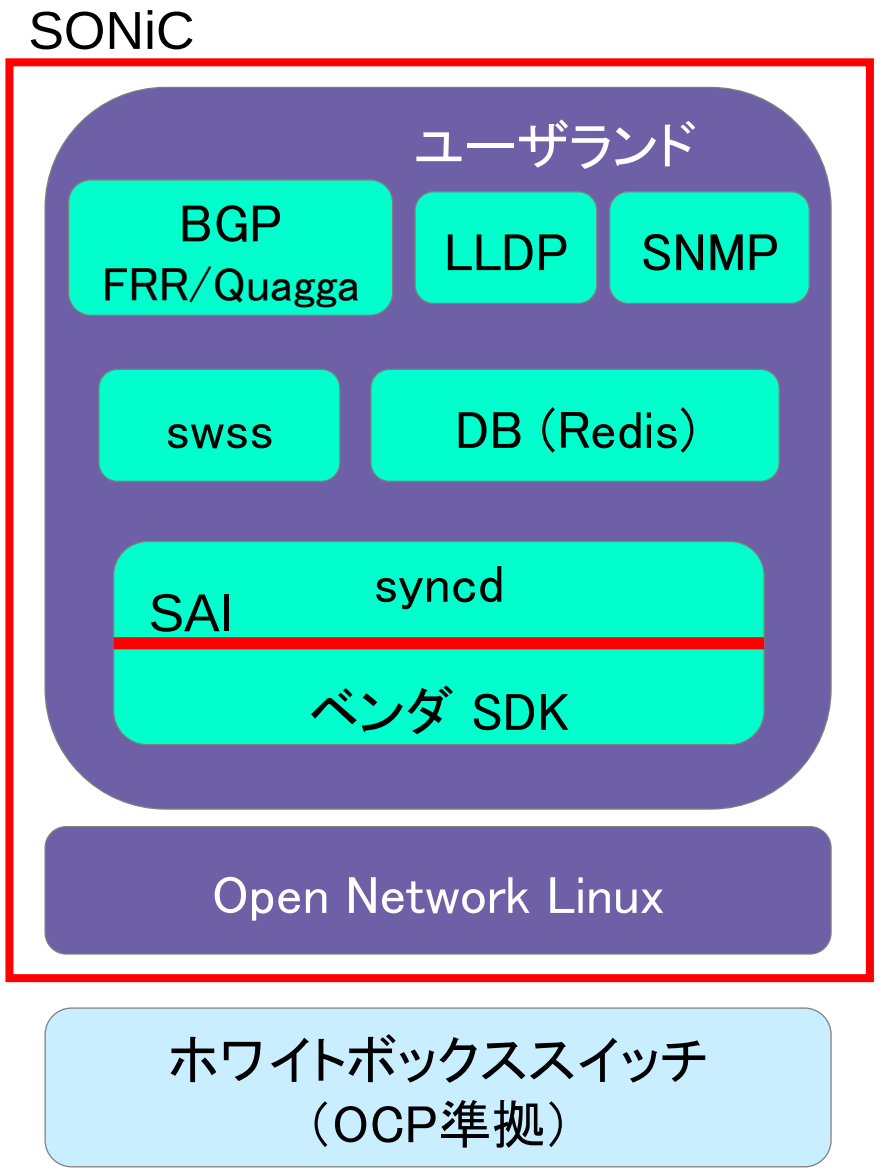
SONiC紹介

APRESIA®

JANOG44

SONiC（Software for Open Networking in the Cloud）とは？

- ホワイトボックススイッチ用のOSSのNOS
  - 厳密にはONL上で動作するソフトウェア群
  - Apache License ver.2
- Microsoftが公開したソースコードが母体
- SAI (Switch Abstraction Interface)を定義し、スイッチチップの差分を隠蔽したことで、マルチベンダ対応を実現
- BGPやLLDP、データベースなどのアプリケーションはコンテナ化
  - 2019/4よりデフォルトのL3スタックとしてFRRoutingを採用
- BGPを使ってIP CLOSファブリックを構築可能



SONiC参加企業・団体

https://azure.github.io/SONiC/

SONiC 対応デバイスの状況

APRESIA

- スイッチベンダ
  - Alpha
  - Arista
  - WNC
  - Edgecore
  - Celestica
  - Dell
  - Delta
  - Embedway
  - Ingrasys
  - Inventec
  - MiTAC
  - Quanta
- 対応チップ
  - Broadcom
  - Barefoot
  - Centec
  - Cavium
  - Marvell
  - Mellanox
  - Nephos

Trident/Tomahawk系だけでなく、
Jericho系統もサポート対象に

Supported Devices and Platforms

Xin Liu edited this page on 2 Apr · 70 revisions

Following is the list of platforms that support SONiC. Last updated Mar 2018.

Switch Vendor	Switch SKU	ASIC Vendor	Swich ASIC	Port Configuration	SONiC Image
Alpha	SNH60B0-640F	Broadcom	Tomahawk2	64x100G	SONiC-ONIE-Broadcom
Alpha	SNH60A0-320FV2	Broadcom	Tomahawk	32x100G	SONiC-ONIE-Broadcom
Arista	7050QX-32	Broadcom	Trident2	32x40G	SONiC-Aboot-Broadcom[2]

https://github.com/Azure/SONiC/wiki/Supported-Devices-and-Platforms

上記サイトからSONiCのイメージも入手可能

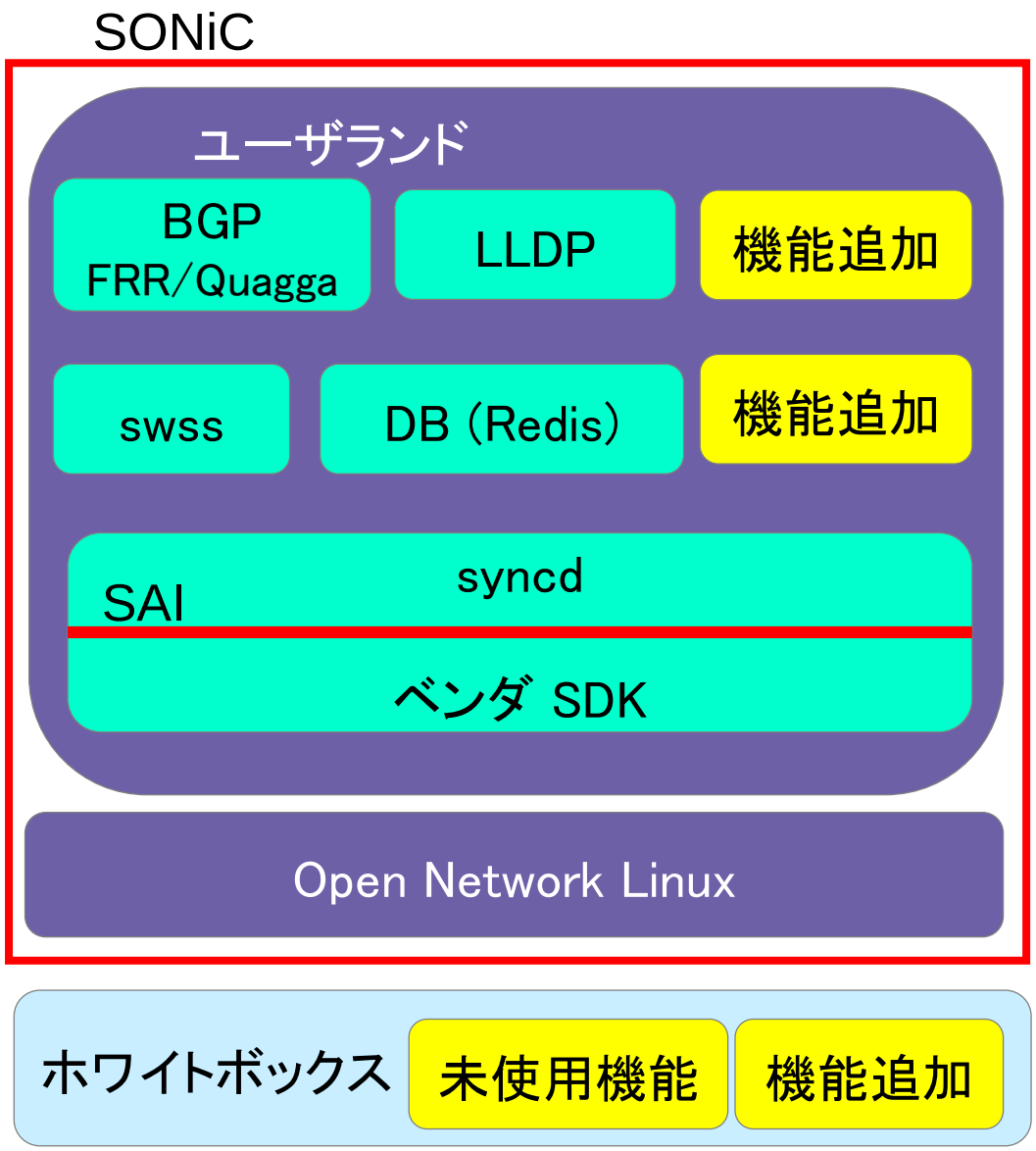
SONiCを調査しているモチベーション

- L2/L3のIP CLOS Fabricの最低限の機能を持ったOSSのNOSであること
- チップ・スイッチベンダがSONiCのサポートに力を割いていること
- 海外では現時点で、商用にて運用している事例が存在すること
  - Microsoft, Alibaba, LinkedIn
- OCP Global SummitにてSONiCのみの独立したWorkshopが開催されるなど、注目が高まっていること
- 一方で、国内での情報が少ないこと
  - 少しでも興味のある方々に有用な情報を纏めたい

OCP Global Summit 2019/3のアジェンダの一部
https://www.opencompute.org/events/past-summits

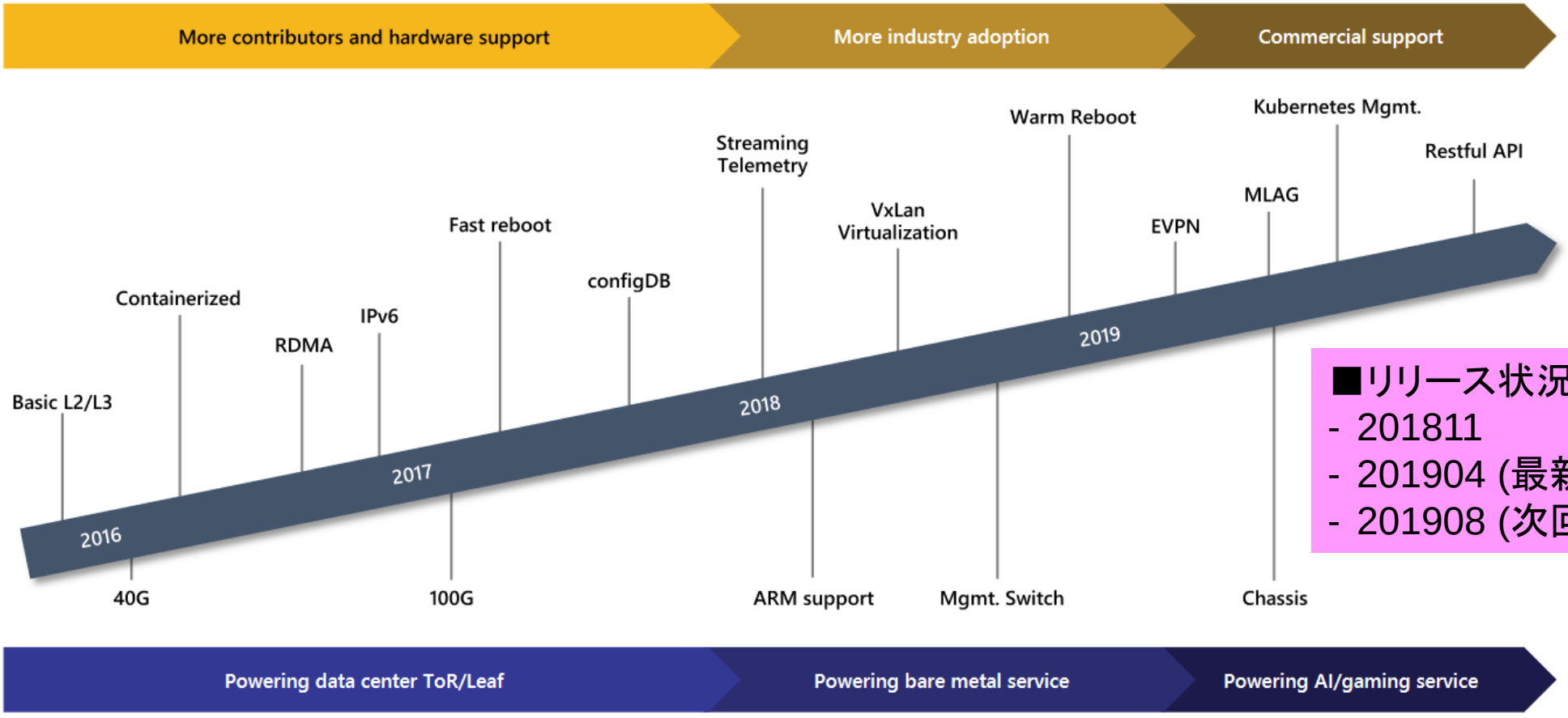
Engineering Workshop: SONiC			
SAI - Updates and Roadmap	Jai Kumar; Guohan Lu	Video	Slides
SONiC Event Driven Dataplane Telemetry	Matty Kadosh; Aviad Raveh	Video	Slides
InBand Network Telemetry an Open and Multi-platform Network Analytics for Disaggregated Data-Centers	Roberto Mari	Video	Slides
Transponder Abstraction Interface (TAI)	Scott Emery; Wataru Ishida	Video	Slides
LinkedIn Adoption of OCP SONiC	Zhenggen Xu	Video	Slides
Advanced Network Telemetry and Analytics Over SONiC	Eli Karpilovski	Video	Slides
Cloud Scale Telemetry – Real-time Telemetry and Analytics at Scale	William Chen; Ashoka Kallappa; Tim Stevenson	Video	Slides
SONiC+: Adding Application Logic to Open NOS	Matty Kadosh; Aviad Raveh	Video	Slides
SONIC/SAI Support in Juniper DC Switches (SONIC / SAI support in Multi-PFE modular chassis) - POC	Lester Bird; Kumuthini Ratnasingham	Video	Slides
Unlocking SONiC's Potential for Intent-Based Data Centers: Three Easy Steps	Nikos Triantafillis	Video	Slides
SONiC + ONL: Open-Source NOS with Broader Platform Support	Wataru Ishida; Steven Noble	Video	Slides
SONiC - Latest Update and Roadmap Forward	Xin Liu; Lihua Yuan	Video	Slides
Alibaba's SONiC Development for Large Scale Deployment and Operation	Guohui Wang	Video	Slides
SONiC Deployments Powered by Programmable Dataplane	Arkadiy Shapiro	Video	Slides
cRPD – Cloud-Grade Routing as a Micro-Service for Open Networking Platforms	Manish Gupta; Vinay Nallamothu	Video	Slides

SONiCを使うことのメリットは？



- 統一NOSによるマルチベンダ対応
- ユーザランドへの機能追加
  - Telemetry等のツール拡張
  - プロトコルスタックの拡張(例：goBGP)
- SONiCベース ＋ データプレーン拡張
  - SONiCでは未使用のハードウェア機能を有効化
  - P4によるデータプレーンプログラムによる機能拡張（INT）
- NOS自体のコスト削減

SONiC roadmap



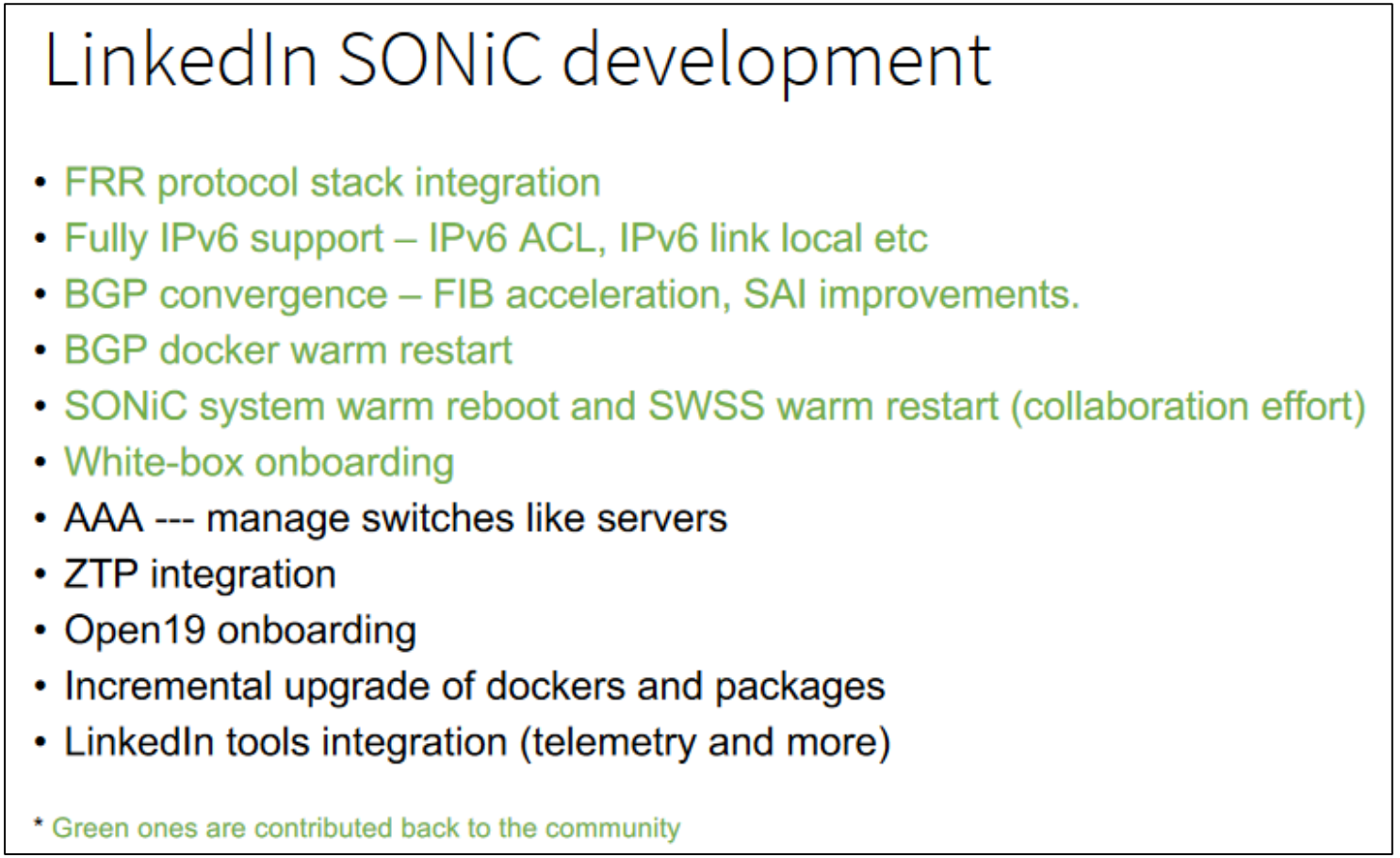
■リリース状況
- 201811
- 201904 (最新)
- 201908 (次回)

2019 OCP Global Summit “SONiC/SAI and It's Rapid Growing Ecosystem”
https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/f86f93bac8c32db39dd851dcbbe5101c64321d91.pdf

ロードマップ詳細は以下を参照
https://github.com/Azure/SONiC/wiki/Sonic-Roadmap-Planning

事例①：LinkedIn

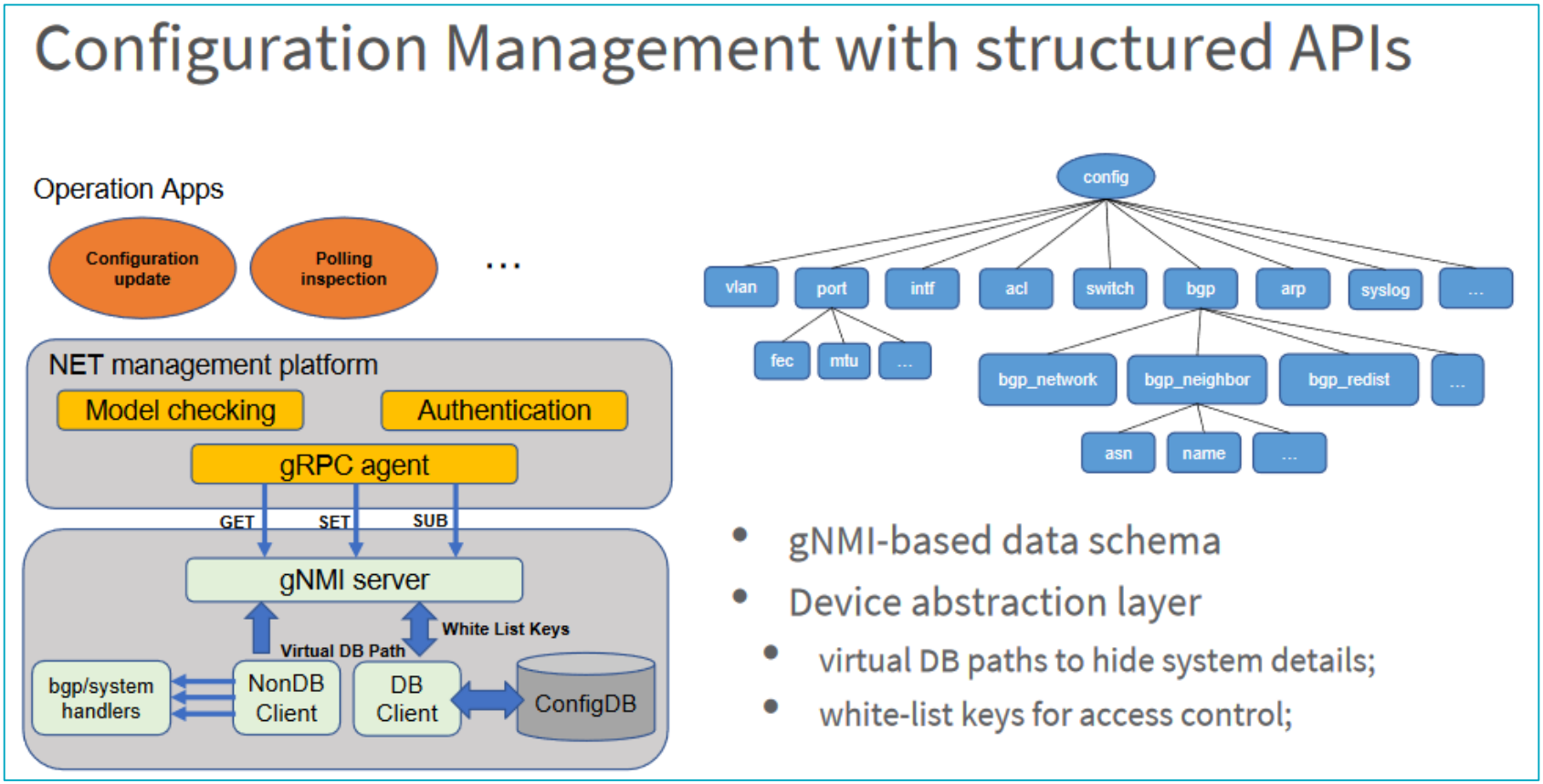
◆SONiC 201904ブランチに取り込まれた機能拡張（緑字）



OCP Global Summit 2019 “LinkedIn Adoption of OCP SONiC”
https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/94fe15451133e4bea215baf9f63a984d624496f9.pdf

事例②： Alibaba (1/2)

◆設定をJson等の構造化されたデータとして扱えることがポイント



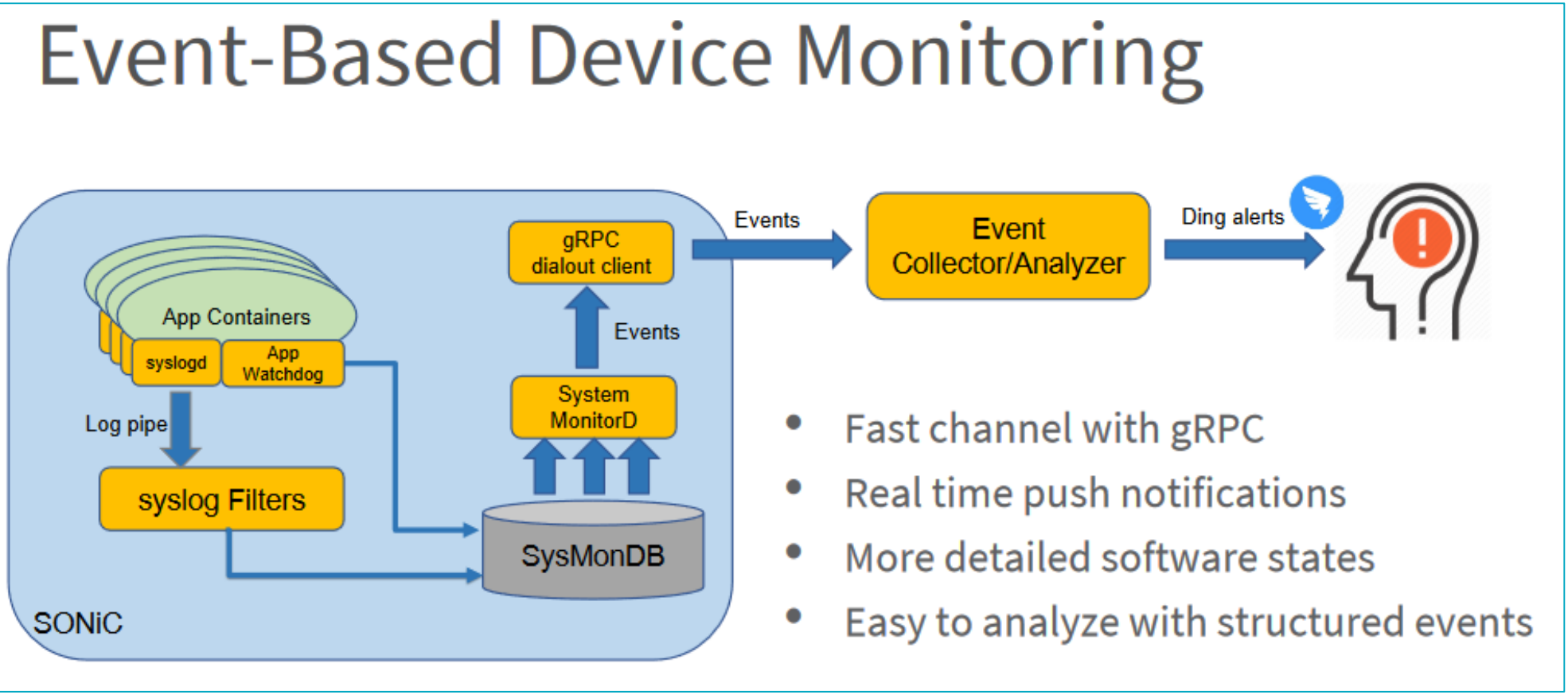
OCP Global Summit 2019
“Alibaba's SONiC Development for Large Scale Deployment and Operation”
https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/aa5200c2e337426ff6c31ced5f2b4b21839a0ba2.pdf

事例②： Alibaba (2/2)

本日簡単なデモをします

APRESIA

◆監視のリアルタイム性向上、構造化されたデータによるイベント通知



OCP Global Summit 2019
“Alibaba's SONiC Development for Large Scale Deployment and Operation”
https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/aa5200c2e337426ff6c31ced5f2b4b21839a0ba2.pdf

SONiCのアーキテクチャ

Telemetry

電源・ファン監視

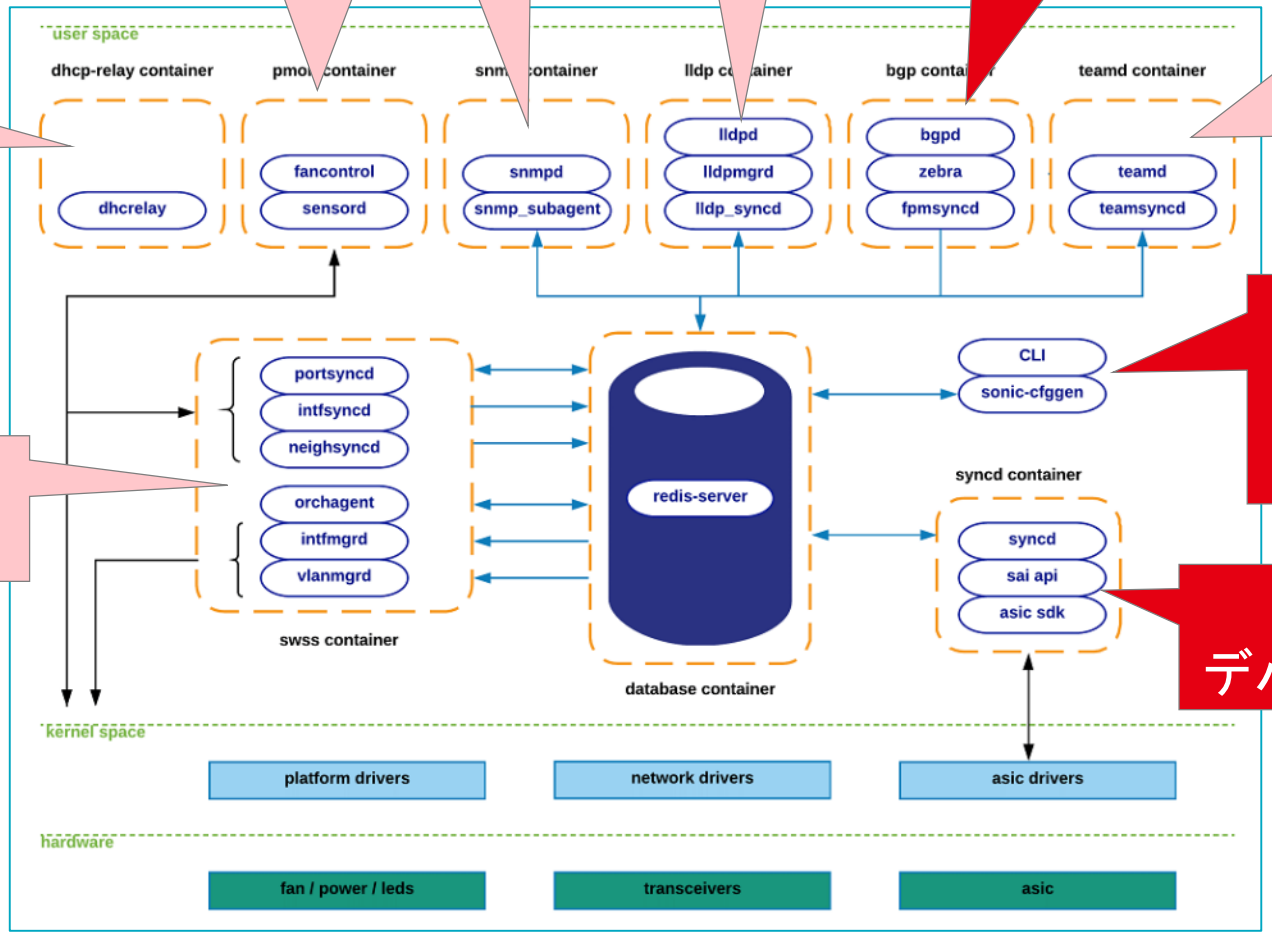
SNMP

LLDP

FRRouting BGP

teamd (LAG) 将来MLAG対応予定

DHCP



CLI config_db (XML)

SWSS オーケストレータ

SAI デバイスドライバ

お話する部分

https://github.com/Azure/SONiC/wiki/Architecture

SONiC仮想マシンによる試験環境構築

APRESIA®

JANOG44

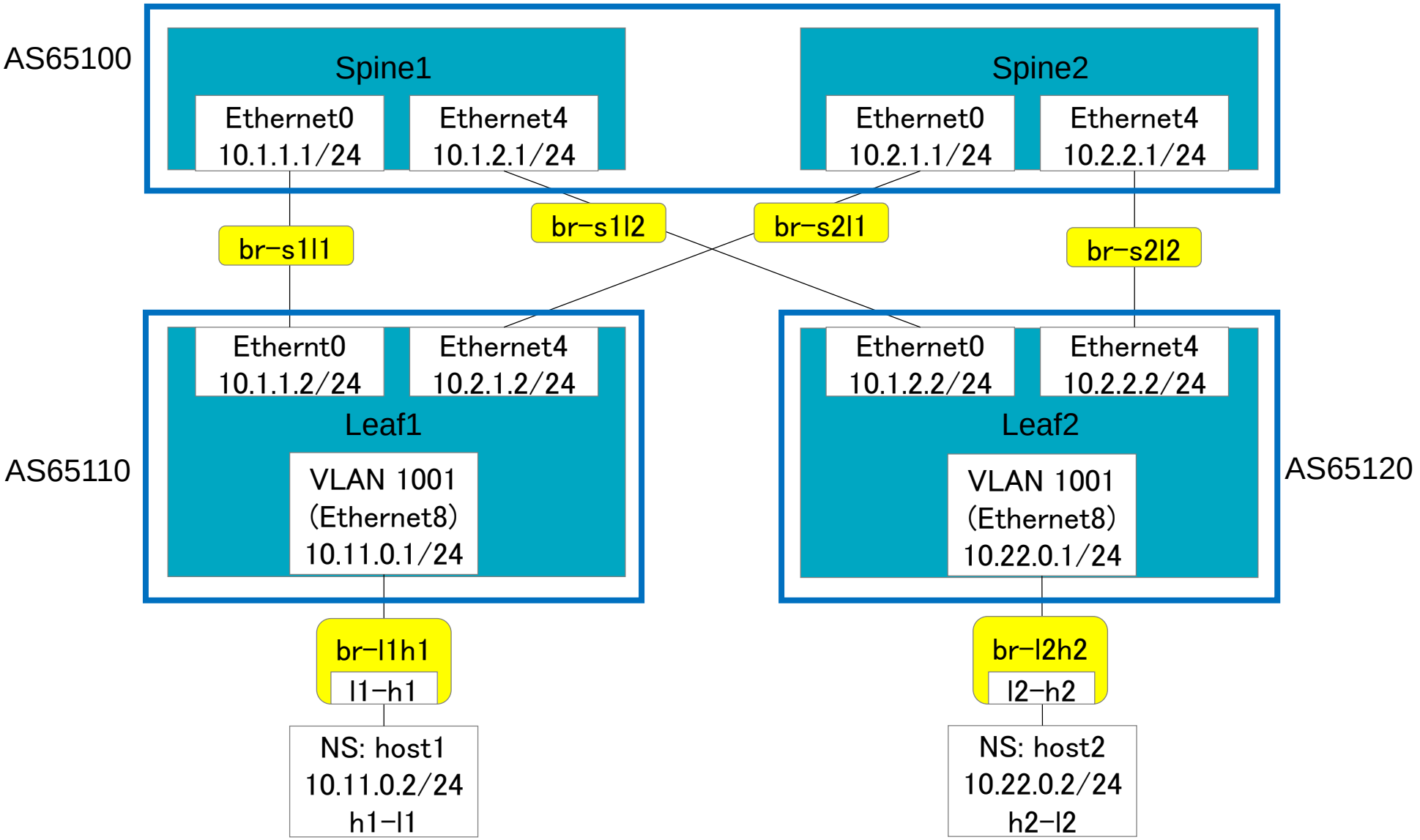
SONiCの仮想化環境での試験方法

- SONiCではvirtual switchのPlatformも用意されている
  - KVMとDockerの環境あり
    - https://github.com/Azure/sonic-buildimage/blob/master/platform/vs/README.vsvm.md
    - https://github.com/Azure/sonic-buildimage/blob/master/platform/vs/README.vsdocker.md
  - DailyビルドされたKVMとDockerのイメージを入手可能
    - https://sonic-jenkins.westus2.cloudapp.azure.com/job/vs/
- 今回の試験方法
  - 201904(最新)ブランチのSONiC KVMイメージにて試験環境を構築
    - Docker版のSONiCでは機能が制限されるため

※環境構築の詳細は以下のテクニカルブログにて公開予定

https://www.apresiatac.jp/blog/

試験構成 SONiC仮想マシン 4台構成



事前準備

- 環境条件
  - 仮想マシンUbuntu18.04 (kernel 4.15.0-54)
    - KVM:　libvirt 4.0.0, QEMU(API) 4.0.0, QEMU(hypervisor) 2.11.1
      - KVMのネストを使うことで、仮想マシン上にてKVMを使用可能
- 必要ファイルの入手
  - 以下から、”sonic-vs.img.gz”をダウンロードし、解凍
    - https://sonic-jenkins.westus2.cloudapp.azure.com/job/vs/job/buildimage-vs-image-201904/12/artifact/target/
  - 以下から、”sonic.xml”をダウンロード
    - https://raw.githubusercontent.com/Azure/sonic-buildimage/201904/platform/vs/sonic.xml
- 以下の4つのディレクトリを用意し、それぞれに上記の2つのファイルを配置
  - spine1, spine2, leaf1, leaf2
- sonic.xmlを環境に合わせて変更

sonic.xmlの例（Spine1）

```
<?xml version="1.0" encoding="utf-8"?>
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <name>sonic-spine1</name>
  <memory unit='KiB'>2048000</memory>
  <currentMemory unit='KiB'>2048000</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-1.5'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='writeback'/>
      <source file='<環境によって値を変更>/spine1/sonic-vs.img'/>
      <target bus='virtio' dev='vda'/>
    </disk>
```

sonic.xmlの例 (Spine1)

```
<serial type='tcp'>
      <source host='127.0.0.1' mode='bind' service='7001'/>
      <target port='0'/>
      <protocol type='telnet'/>
    </serial>
    <interface type='network'>
      <source network='default'/>
      <model type='virtio'/>
    </interface>
    <interface type='bridge'>
      <source bridge='br-sll1'/>
      <model type='virtio'/>
    </interface>
    <interface type='bridge'>
      <source bridge='br-sll2'/>
      <model type='virtio'/>
    </interface>
    <controller type='usb' index='0'/>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='apparmor' relabel='yes'/>
</domain>
```

シリアル(Telnet接続)

一つ目のInterfaceは管理用インターフェース (eth0)

Ethernet0 (eth1)

Ethernet4 (eth2)

SONiC仮想マシンの起動

- virshコマンドで各装置を起動 (例: spine1、ファイル場所は適宜読み替え)
  - virsh create spine1/sonic.xml
- 以下のようにSONiCの仮想マシンが起動

```
$ virsh list
 Id    Name                           State
----------------------------------------------------
 96    sonic-spine1                   running
 97    sonic-spine2                   running
 98    sonic-leaf1                    running
 99    sonic-leaf2                    running
```

- sonic.xmlに記載したポート番号に対してtelnetすればログイン可能
  - ユーザ:　admin
  - パスワード: YourPaSsWoRd

SONiCへログイン（例:Spine1）

```
$ telnet 127.0.0.1 7001
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.

Debian GNU/Linux 9 Spine1 ttyS0

Spine1 login: admin
Password:
Last login: Mon Jun 24 18:33:03 UTC 2019 from 10.0.0.2 on pts/0
Linux Spine1 4.9.0-9-2-amd64 #1 SMP Debian 4.9.168-1+deb9u2 (2015-12-19) x86_64
You are on
  ____   ___ _   _ _  ____
 / ___| / _ ¥| ¥ | (_)/ ___|
 ¥___ ¥| | | |  ¥| | | |
  ___) | |_| | |¥  | | |___
 |____/ ¥___/|_| ¥_|_|¥____|

-- Software for Open Networking in the Cloud --

Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:    http://azure.github.io/SONiC/

$
```

SONiCのコマンドライン： config系

```
admin@Spine1:~$ config ?
Usage: config [OPTIONS] COMMAND [ARGS]...

  SONiC command line - 'config' command

Options:
  --help  Show this message and exit.

Commands:
  aaa                     AAA command line
  acl                     ACL-related configuration tasks
  bgp                     BGP-related configuration tasks
  ecn                     ECN-related configuration tasks
  interface               Interface-related configuration tasks
  interface_naming_mode   Modify interface naming mode for interacting...
  load                    Import a previous saved config DB dump file.
  load_mgmt_config        Reconfigure hostname and mgmt interface based...
  load_minigraph          Reconfigure based on minigraph.
  mirror_session
  platform                Platform-related configuration tasks
  portchannel
  qos
  reload                  Clear current configuration and import a...
  save                    Export current config DB to a file on disk.
  tacacs                  TACACS+ server configuration
  vlan                    VLAN-related configuration tasks
(省略)
```

SONiCのコマンドライン： show系

```
admin@Spine1:~$ show ?
Usage: show [OPTIONS] COMMAND [ARGS]...

  SONiC command line - 'show' command

Options:
  -?, -h, --help  Show this message and exit.

Commands:
  aaa                     Show AAA configuration
  acl                     Show ACL related information
  arp                     Show IP ARP table
  bgp                     BGP information
  clock                   Show date and time
  ecn                     Show ECN configuration
  environment             Show environmentals (voltages, fans, temps)
  interfaces              Show details of the network interfaces
  ip                      Show IP (IPv4) commands
  ipv6                    Show IPv6 commands
  line                    Show all /dev/ttyUSB lines and their info
  lldp                    LLDP (Link Layer Discovery Protocol)...
  logging                 Show system log
  mac                     Show MAC (FDB) entries
  mirror_session          Show existing everflow sessions
  mmu                     Show mmu configuration
 (省略)
```

SONiCの構成情報の考え方

- SONiCのコマンドリファレンスは以下を参照
  - https://github.com/Azure/sonic-utilities/blob/master/doc/Command-Reference.md
- SONiCのスタートアップconfig
  - “/etc/sonic/config_db.json”に、json形式でconfigを記述（以下を参照）
    - https://github.com/Azure/SONiC/wiki/Configuration
  - rebootあるいはconfig reloadにて設定を反映

config_db.jsonの例 (leaf1)

```
{
    "BGP_NEIGHBOR": {
        "10.1.1.1": {
            "asn": "65100",
            "holdtime": "180",
            "keepalive": "60",
            "local_addr": "10.1.1.2",
            "name": "Spine1",
            "nhopself": "0",
            "rrclient": "0"
        },
        "10.2.1.1": {
            "asn": "65100",
            "holdtime": "180",
            "keepalive": "60",
            "local_addr": "10.2.1.2",
            "name": "Spine2",
            "nhopself": "0",
            "rrclient": "0"
        }
    },
```

BGP Neighbor設定

```
"DEVICE_METADATA": {
        "localhost": {
            "bgp_asn": "65110",
            "hostname": "Leaf1",
            "hwsku": "Force10-S6000",
            "mac": "52:54:00:12:34:01",
            "platform": "x86_64-kvm_x86_64-r0",
            "type": "LeafRouter"
        }
    },
    "INTERFACE": {
        "Ethernet0|10.1.1.2/24": {},
        "Ethernet4|10.2.1.2/24": {}
    },
    "VLAN_INTERFACE": {
        "Vlan1001|10.11.0.1/24": {}
    },
    "LOOPBACK_INTERFACE": {
        "Loopback0|3.3.3.3/32": {}
    },
    "PORT": {
        "Ethernet0": {
            "admin_status": "up",
            "alias": "fortyGigE0/0",
            "index": "0",
(省略)
```

自装置設定

MACはKVM毎に値を変えておくこと

IPアドレス 物理ポート

IPアドレス VLAN

loopbackアドレス Router IDに使用

物理ポート設定

config_db.jsonの例 (leaf1)

```
(続き)
            "mtu": "9100",
            "speed": "40000"
        }
    },
    "TELEMETRY": {
        "gnmi": {
            "port": "8080"
        }
    },
    "VLAN": {
        "Vlan1001": {
            "members": [
                "Ethernet8"
            ],
            "vlanid": "1001"
        }
    },
    "VLAN_MEMBER": {
        "Vlan1001|Ethernet8": {
            "tagging_mode": "untagged"
        }
    }
}
```

VLAN設定

config_db.jsonからFRRの設定に反映

- “sonic-cfggen”よりconfig_db.jsonを読み込み、各コンテナに必要な情報をインプット
- bgpコンテナ内の”/usr/share/sonic/templates/bgpd.conf.j2”のjinja2のテンプレートから、FRR用の設定(/etc/frr/bgpd.conf)が生成される

  ※BGPのpeeringが開始されない場合は” sudo config bgp startup all”を実行

```
root@Leaf1:/# cat /usr/share/sonic/templates/bgpd.conf.j2
!
 (中略)
!
router bgp {{ DEVICE_METADATA['localhost']['bgp_asn'] }}
  bgp log-neighbor-changes
  bgp bestpath as-path multipath-relax
  no bgp default ipv4-unicast
  bgp graceful-restart restart-time 240
  bgp graceful-restart
{% if DEVICE_METADATA['localhost']['type'] == 'ToRRouter' %}
  bgp graceful-restart preserve-fw-state
{% endif %}
{% for (name, prefix) in LOOPBACK_INTERFACE|pfx_filter %}
{% if prefix | ipv4 and name == 'Loopback0' %}
  bgp router-id {{ prefix | ip }}
{% endif %}
{% endfor %}
{# advertise loopback #}
 (省略)
```

```
root@Leaf1:/# cat /etc/frr/bgpd.conf
!
 (中略)
!
router bgp 65110
  bgp log-neighbor-changes
  bgp bestpath as-path multipath-relax
  no bgp default ipv4-unicast
  bgp graceful-restart restart-time 240
  bgp graceful-restart
  bgp router-id 3.3.3.3
  network 3.3.3.3/32
  network 10.11.0.1/24
  neighbor 10.1.1.1 remote-as 65100
  neighbor 10.1.1.1 description Spine1
  address-family ipv4
    neighbor 10.1.1.1 activate
    neighbor 10.1.1.1 soft-reconfiguration inbound
    maximum-paths 64
 (省略)
```

“bgpd.conf.j2”は以下にて参照可能
https://github.com/Azure/sonic-buildimage/blob/201904/dockers/docker-fpm-frr/bgpd.conf.j2

config_db.jsonのFFRへの反映 (leaf1)

```
$ vtysh

Hello, this is FRRouting (version 7.0.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

Leaf1# show running-config
Building configuration...

Current configuration:
!
frr version 7.0.1-sonic
frr defaults traditional
hostname Leaf1
log syslog informational
log facility local4
no service integrated-vtysh-config
!
password zebra
enable password zebra
!
router bgp 65110
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 bgp graceful-restart restart-time 240
 bgp graceful-restart
 bgp bestpath as-path multipath-relax
 neighbor 10.1.1.1 remote-as 65100
 neighbor 10.1.1.1 description Spine1
 neighbor 10.2.1.1 remote-as 65100
 neighbor 10.2.1.1 description Spine2
!
```

```
 address-family ipv4 unicast
  network 3.3.3.3/32
  network 10.11.0.0/24
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 soft-reconfiguration inbound
  neighbor 10.2.1.1 activate
  neighbor 10.2.1.1 soft-reconfiguration inbound
  maximum-paths 64
 exit-address-family
!
route-map RM_SET_SRC permit 10
 set src 3.3.3.3
!
route-map set-next-hop-global-v6 permit 10
 set ipv6 next-hop prefer-global
!
route-map ISOLATE permit 10
 set as-path prepend 65110
!
route-map TO_BGP_SPEAKER_V4 deny 10
!
route-map FROM_BGP_SPEAKER_V4 permit 10
!
ip protocol bgp route-map RM_SET_SRC
!
line vty
!
end
```

Leaf1, 2のBGPルーティング状況を確認

APRESIA

◆Leaf 1, 2の配下のhostネットワークがBGPによって広告されている

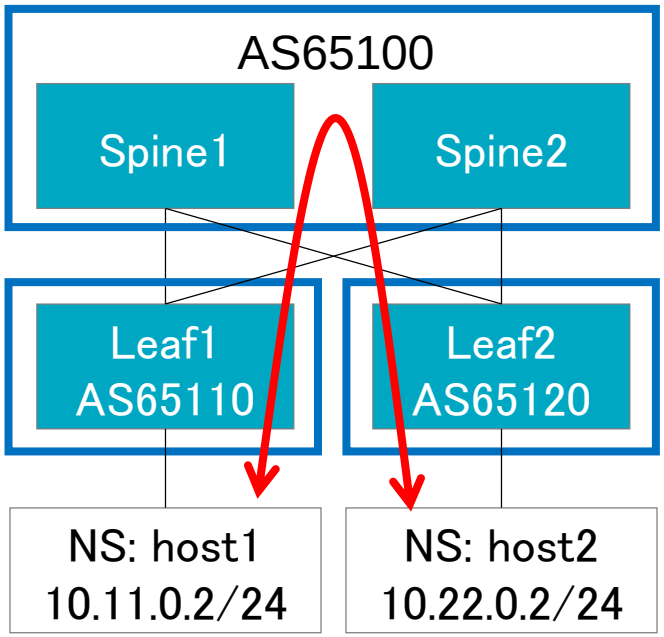
```
Leaf1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route

B>* 1.1.1.1/32 [20/0] via 10.1.1.1, Ethernet0, 00:03:17
  *                   via 10.2.1.1, Ethernet4, 00:03:17
B>* 2.2.2.2/32 [20/0] via 10.1.1.1, Ethernet0, 00:03:17
  *                   via 10.2.1.1, Ethernet4, 00:03:17
C>* 3.3.3.3/32 is directly connected, lo, 00:03:59
B>* 4.4.4.4/32 [20/0] via 10.1.1.1, Ethernet0, 00:01:39
  *                   via 10.2.1.1, Ethernet4, 00:01:39
C>* 10.1.1.0/24 is directly connected, Ethernet0, 00:03:19
B>* 10.1.2.0/24 [20/0] via 10.1.1.1, Ethernet0, 00:01:39
  *                    via 10.2.1.1, Ethernet4, 00:01:39
C>* 10.2.1.0/24 is directly connected, Ethernet4, 00:03:19
B>* 10.2.2.0/24 [20/0] via 10.1.1.1, Ethernet0, 00:01:39
  *                    via 10.2.1.1, Ethernet4, 00:01:39
C>* 10.11.0.0/24 is directly connected, Ethernet8, 00:03:19
B>* 10.22.0.0/24 [20/0] via 10.1.1.1, Ethernet0, 00:01:39
  *                     via 10.2.1.1, Ethernet4, 00:01:39
Leaf1#
```

```
Leaf2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route

B>* 1.1.1.1/32 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                   via 10.2.2.1, Ethernet4, 00:02:44
B>* 2.2.2.2/32 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                   via 10.2.2.1, Ethernet4, 00:02:44
B>* 3.3.3.3/32 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                   via 10.2.2.1, Ethernet4, 00:02:44
C>* 4.4.4.4/32 is directly connected, lo, 00:03:24
B>* 10.1.1.0/24 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                    via 10.2.2.1, Ethernet4, 00:02:44
C>* 10.1.2.0/24 is directly connected, Ethernet0, 00:02:48
B>* 10.2.1.0/24 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                    via 10.2.2.1, Ethernet4, 00:02:44
C>* 10.2.2.0/24 is directly connected, Ethernet4, 00:02:47
B>* 10.11.0.0/24 [20/0] via 10.1.2.1, Ethernet0, 00:02:44
  *                     via 10.2.2.1, Ethernet4, 00:02:44
C>* 10.22.0.0/24 is directly connected, Ethernet8, 00:02:46
Leaf2#
```

Host1, 2間の通信確認

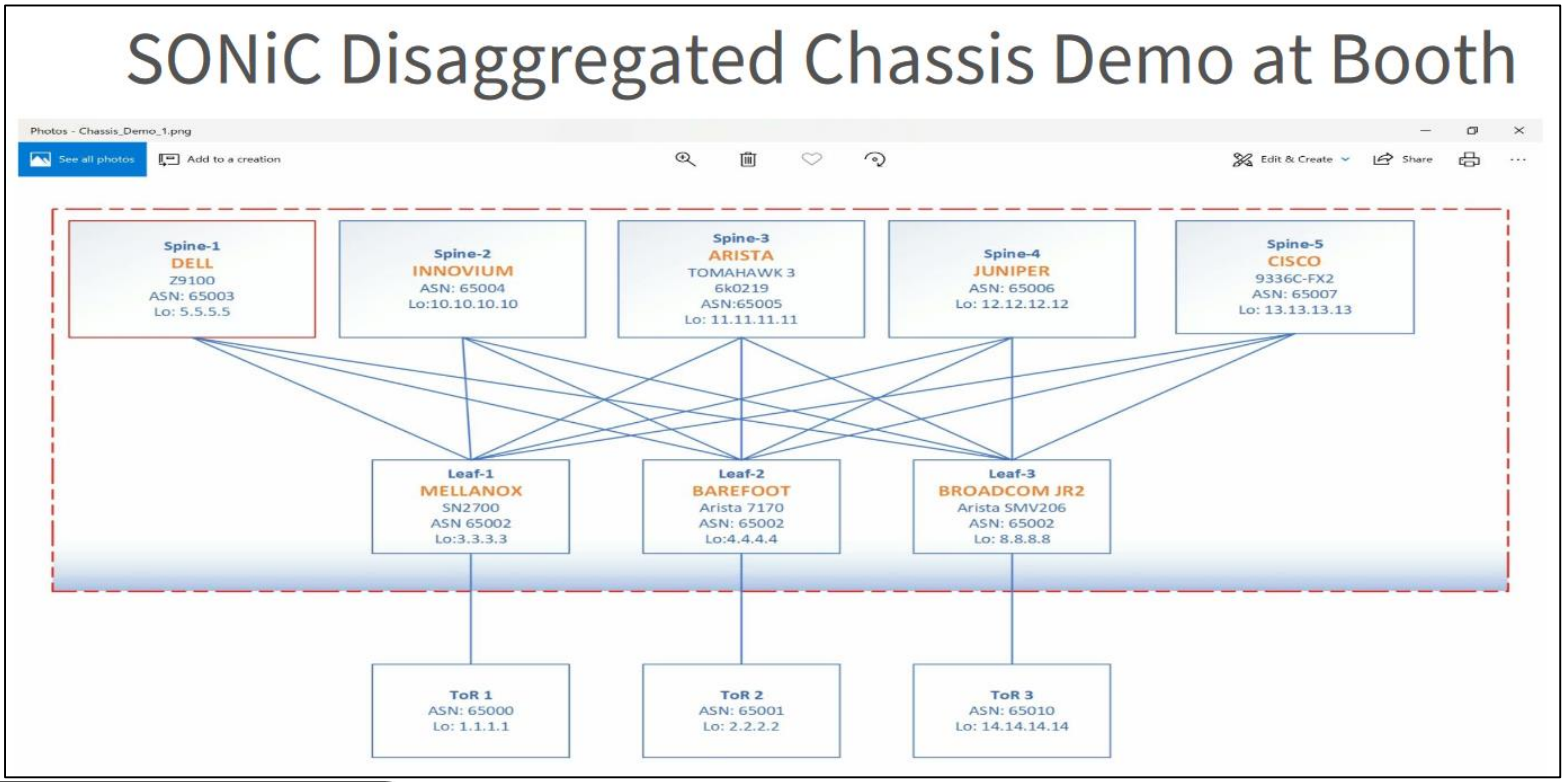



```
$ sudo ip netns exec host1 ping 10.22.0.2
PING 10.22.0.2 (10.22.0.2) 56(84) bytes of data.
64 bytes from 10.22.0.2: icmp_seq=1 ttl=61 time=1.08 ms
64 bytes from 10.22.0.2: icmp_seq=2 ttl=61 time=4.71 ms
64 bytes from 10.22.0.2: icmp_seq=3 ttl=61 time=0.936 ms
64 bytes from 10.22.0.2: icmp_seq=4 ttl=61 time=1.13 ms
64 bytes from 10.22.0.2: icmp_seq=5 ttl=61 time=1.32 ms
64 bytes from 10.22.0.2: icmp_seq=6 ttl=61 time=1.06 ms
^C
--- 10.22.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5037ms
rtt min/avg/max/mdev = 0.936/1.711/4.711/1.347 ms

$ sudo ip netns exec host2 ping 10.11.0.2
PING 10.11.0.2 (10.11.0.2) 56(84) bytes of data.
64 bytes from 10.11.0.2: icmp_seq=1 ttl=61 time=1.56 ms
64 bytes from 10.11.0.2: icmp_seq=2 ttl=61 time=1.02 ms
64 bytes from 10.11.0.2: icmp_seq=3 ttl=61 time=1.13 ms
64 bytes from 10.11.0.2: icmp_seq=4 ttl=61 time=1.14 ms
64 bytes from 10.11.0.2: icmp_seq=5 ttl=61 time=1.20 ms
^C
--- 10.11.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.023/1.214/1.569/0.188 ms
```

今後SONiCにて対応されると思われる機能

- BGP unnumbered + RFC5549　※トライしている人たちもいるが動作未確認
- EVPN VXLAN
- MLAG (サーバ収容の装置冗長)
- Virtual chassis （下図）

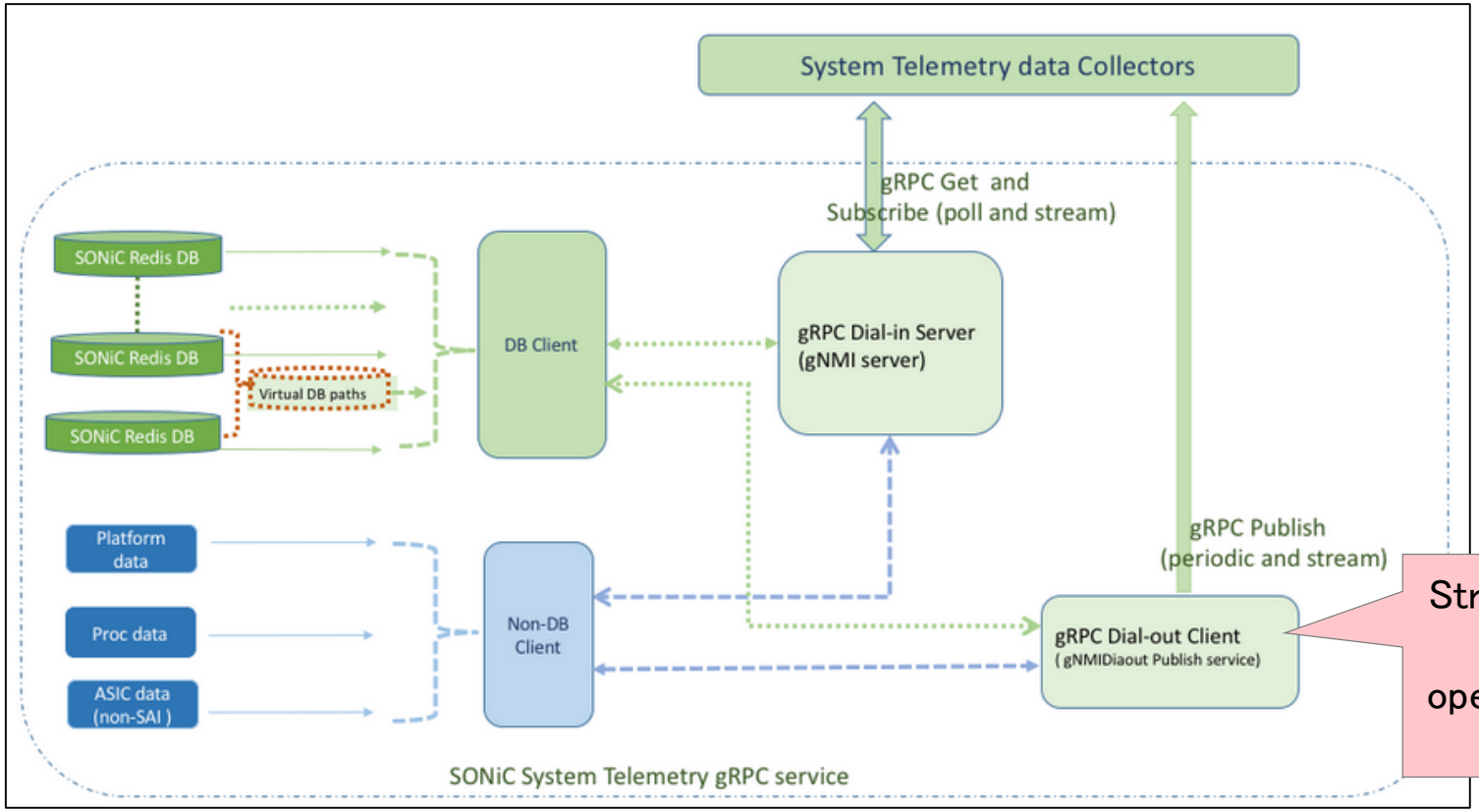


OCP global summit 2019/3
https://146a55aca6f00848c565-a7635525d40ac1c70300198708936b4e.ssl.cf1.rackcdn.com/images/69b40fc6b426e4580dbeb7ff3ded8b64b4294fd2.pdf

SONiCのStreaming Telemetry

◆環境準備など、以下を参照

- ◇ https://github.com/Azure/sonic-telemetry
- ◇ https://github.com/Azure/sonic-telemetry/blob/master/doc/grpc_telemetry.md
- ◇ https://github.com/Azure/sonic-telemetry/blob/master/doc/dialout.md



Streaming Telemetry based on openconfig telemetry yang model

Streaming Telemetry環境の構築

- ツールのインストール
  - SONIC
    - go get -u github.com/Azure/sonic-telemetry/dialout/dialout_client_cli
  - Telemetry Collector側
    - go get -u github.com/Azure/sonic-telemetry/dialout/dialout_server_cli
- ツールの実行
  - Telemetry Collector側
    - ./dialout_server_cli -allow_no_client_auth -logtostderr -port 8081 -insecure -v 2
  - SONIC
    - Config_db.jsonにStreaming Telemetryを設定した上で以下を実行
    - /usr/sbin/dialout_client_cli -insecure -logtostderr -v 1

config_db.json内のStreaming Telemetry設定

```
"TELEMETRY_CLIENT": {
    "Global": {
        "encoding": "JSON_IETF",
        "retry_interval": "30",
        "src_ip": "192.168.122.180",
        "unidirectional": "true"
    },
    "DestinationGroup_HS": {
        "dst_addr": "192.168.122.1:8081"
    },
    "Subscription_HS_RDMA": {
        "dst_group": "HS",
        "path_target": "COUNTERS_DB",
        "paths": "COUNTERS/Ethernet*,COUNTERS_PORT_NAME_MAP",
        "report_interval": "5000",
        "report_type": "periodic"
    }
}
```

Encodingは他に”ASCII”, “BYTES”, “PROTO”を選択可能

送信元のSONiCのIPアドレス

送信先のTelemetryのコレクタのIPアドレス

Streaming Telemetry実施例

■ SONiC側

```
admin@Leaf1:~/telemetry/bin$ ./dialout_client_cli  -insecure -logtostderr -v 1
I0713 11:02:44.020514    9242 dialout_client_cli.go:43] Starting telemetry publish client
I0713 11:02:44.029522    9242 dialout_client.go:675] psubscribe to __keyspace@4__:TELEMETRY_CLIENT|* failed dial unix
/var/run/redis/redis.sock: connect: permission denied
I0713 11:02:44.042284    9242 dialout_client_cli.go:45] Exiting telemetry publish client: psubscribe to __keyspac ./dialout_clie
nt_cli  -insecure -logtostderr -v 1dialout_client_cli  -insecure -logtostderr -v 1                    d./dialout_client
I0713 11:02:47.170522    9250 dialout_client_cli.go:43] Starting telemetry publish client
I0713 11:02:47.191440    9250 virtual_db.go:142] PFC WD not enabled on device
I0713 11:02:47.225113    9250 dialout_client.go:318] Dialout service connected to {192.168.122.1:8081} successfully for HS_RDMA
```

■ collector側

```
$ ./dialout_server_cli -allow_no_client_auth -logtostderr -port 8081 -insecure -v 2
I0713 20:05:22.479318    3325 dialout_server.go:66] Created Server on localhost:8081
I0713 20:05:22.479409    3325 dialout_server_cli.go:90] Starting RPC server on address: localhost:8081
== subscribeResponse:
update: <
  timestamp: 1563015922616359917
  prefix: <
    target: "COUNTERS_DB"
  >
  update: <
    path: <
      elem: <
        name: "COUNTERS"
      >
      elem: <
        name: "Ethernet*"
      >
```

SONiC実機による

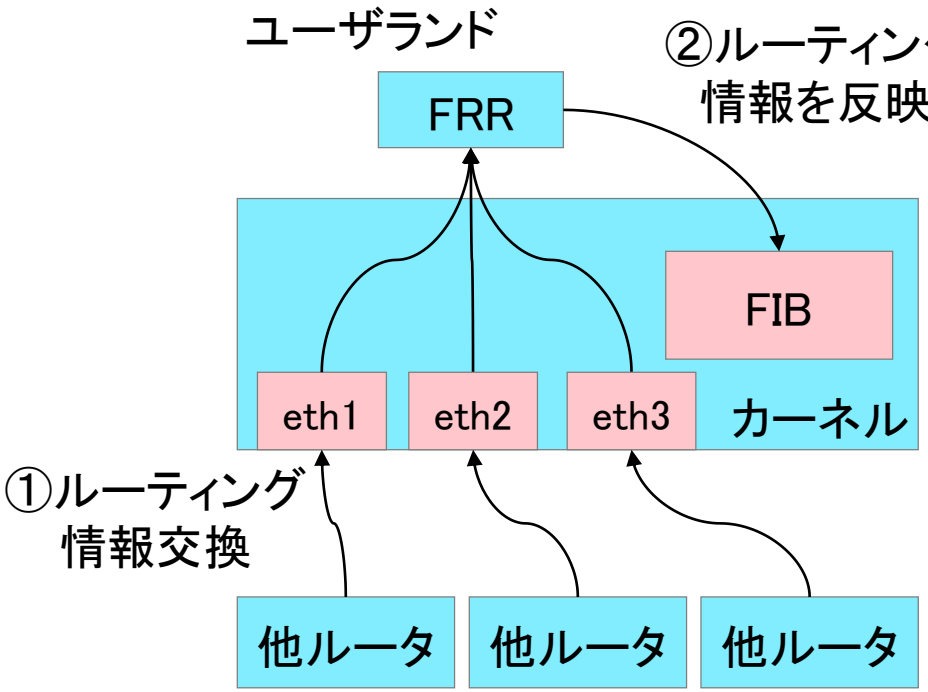
試験環境構築

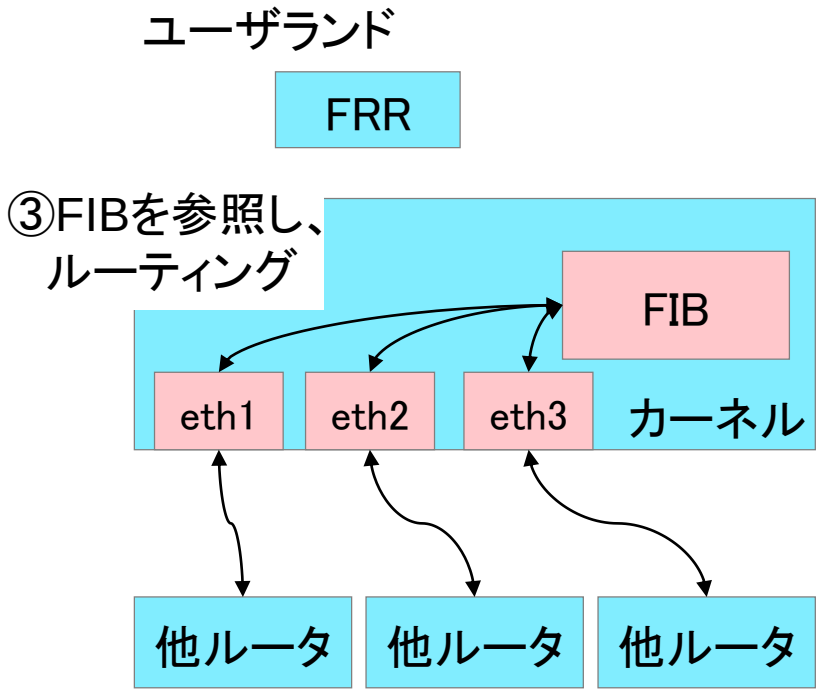
APRESIA®

JANOG44

FRRがLinux上で動作するときのイメージ

- Linuxのカーネルにてネットワークインタフェースを用意
  - FRRはこのインタフェースを介して他ルータとルーティング情報を交換
- FRRが構築したルーティング情報をFIBに登録
- ルーティング実施時は、Linuxカーネルにて FIBを参照





FRRをLinux上で動作させたときのFIBの例

```
:~$ ifconfig -a
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.11  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::eeb:d2ff:fefc:2900  prefixlen 64  scopeid 0x20<link>
        ether 0c:eb:d2:fc:29:00  txqueuelen 1000  (Ethernet)

ens4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fd00:0:0:18::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::eeb:d2ff:fefc:2901  prefixlen 64  scopeid 0x20<link>
        ether 0c:eb:d2:fc:29:01  txqueuelen 1000  (Ethernet)
… (省略)

:~$ ip -6 route show
fd00:0:0:1:1::/80 dev ens6 proto kernel metric 256 pref medium
fd00:0:0:1:2::/80 dev ens7 proto kernel metric 256 pref medium
fd00:0:0:2:1::/80 proto bgp metric 20
        nexthop via fe80::eeb:d2ff:fe81:9f02 dev ens5 weight 1
        nexthop via fe80::eeb:d2ff:fe75:2302 dev ens4 weight 1
…
```

実機（WedgeBF32）にてSONiC起動時のインタフェースの見え方

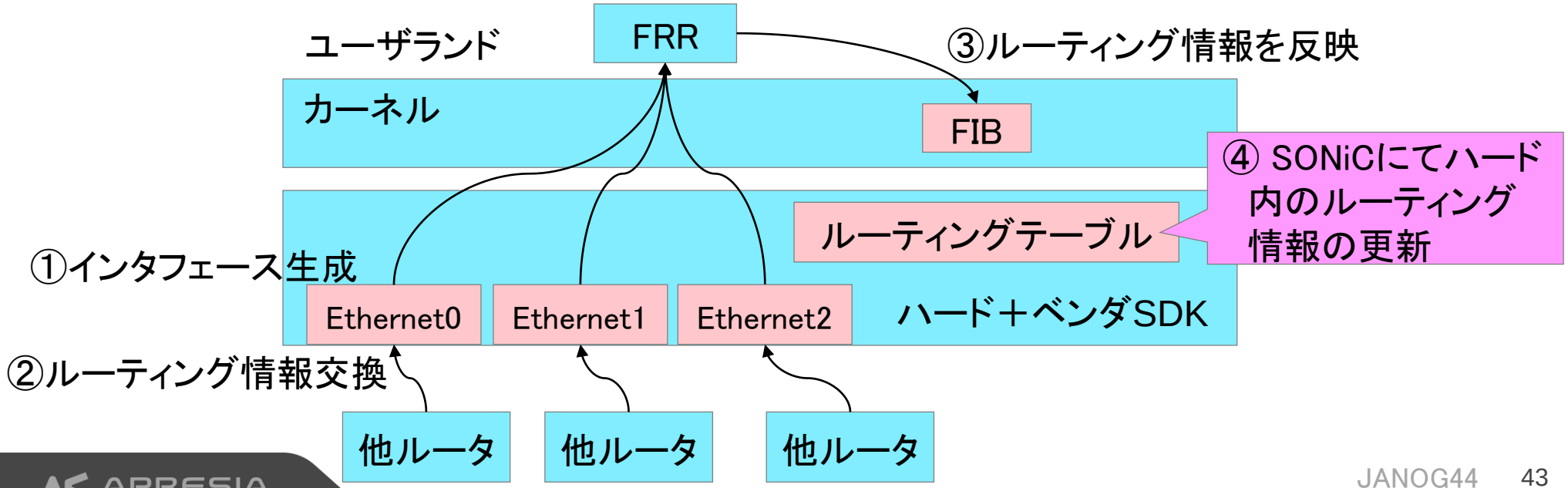
- 物理インタフェースがLinuxのインタフェースとしても見えている
  - これはスイッチ・チップベンダのドライバで実現（SONiCの機能ではない）

```
admin@sonic:~$ show interfaces status
  Interface            Lanes    Speed    MTU       Alias    Vlan    Oper    Admin    Type    Asym PFC
-----------  ---------------  -------  -----  ----------  ------  ------  -------  ------  ----------
  Ethernet0         0,1,2,3     100G   9100   Ethernet0  routed    down       up     N/A         N/A
  Ethernet4         4,5,6,7     100G   9100   Ethernet4  routed    down       up     N/A         N/A
  Ethernet8         8,9,10,11   100G   9100   Ethernet8  routed    down       up     N/A         N/A
 Ethernet12      12,13,14,15    100G   9100  Ethernet12  routed    down       up     N/A         N/A
...

admin@sonic:~$ ip addr
 (中略)
6: Ethernet0: <NO-CARRIER,BROADCAST,UP> mtu 9100 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 00:90:fb:62:0c:57 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.0/31 scope global Ethernet0
       valid_lft forever preferred_lft forever
7: Ethernet4: <NO-CARRIER,BROADCAST,UP> mtu 9100 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 00:90:fb:62:0c:57 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/31 scope global Ethernet4
       valid_lft forever preferred_lft forever
...
```

ホワイボックススイッチ上にてFRRが動作するための課題

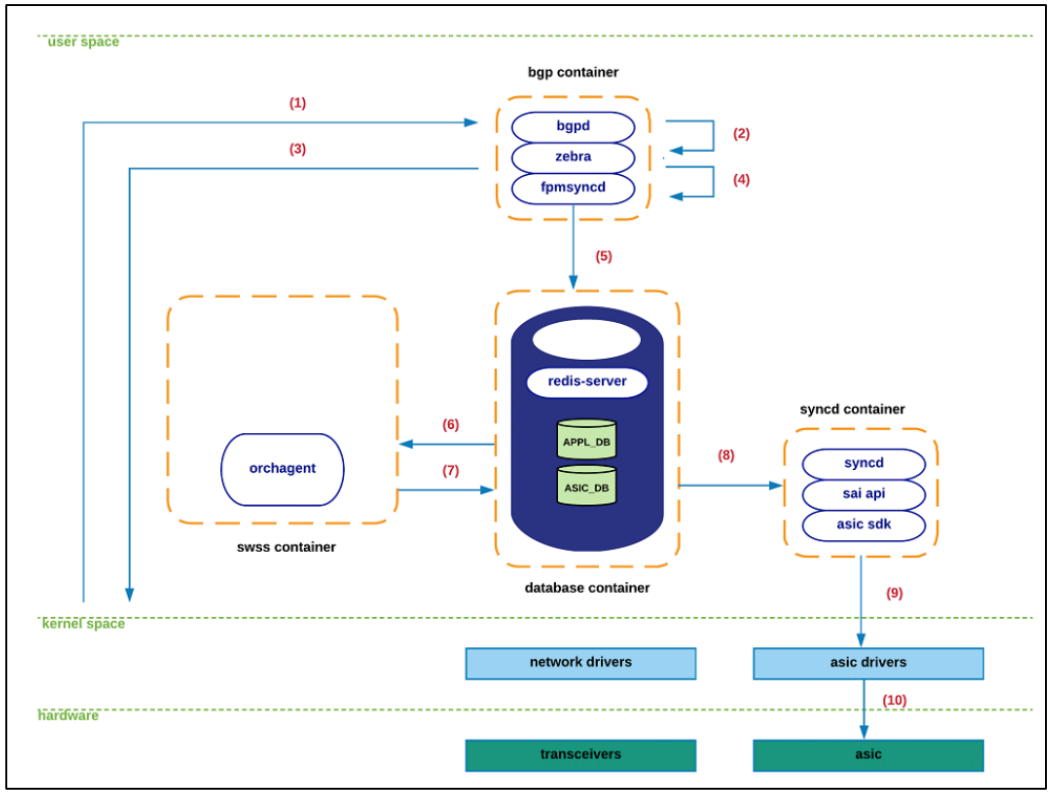
- SONiCが無くても、以下は実施可能
  - ① 物理インタフェースをLinux上のインタフェースとして認識
  - ② 外部ルータと物理インタフェースを介してルーティング情報を交換
  - ③ Linuxカーネル内のFIBテーブルを更新
- SONiCが実施する処理
  - ④ ハードウェア内のルーティングテーブルへの反映

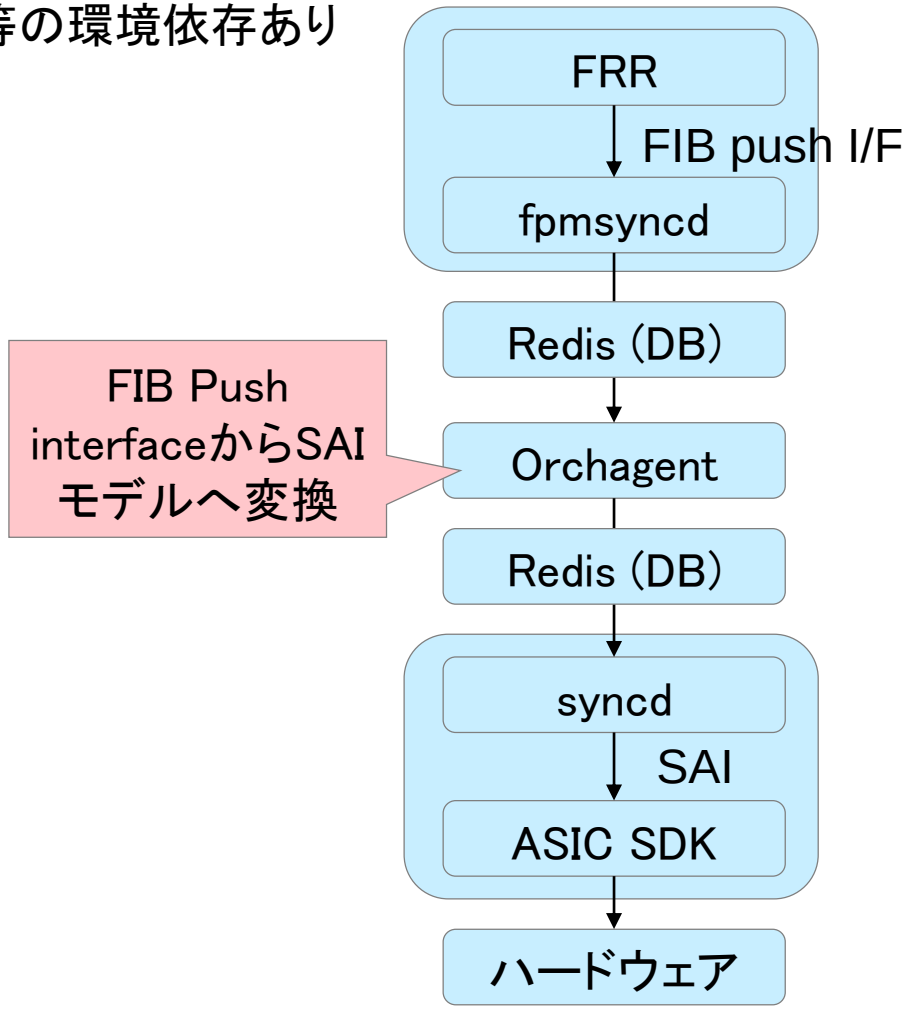


SONiCにてハードウェアルーティングを設定する流れ

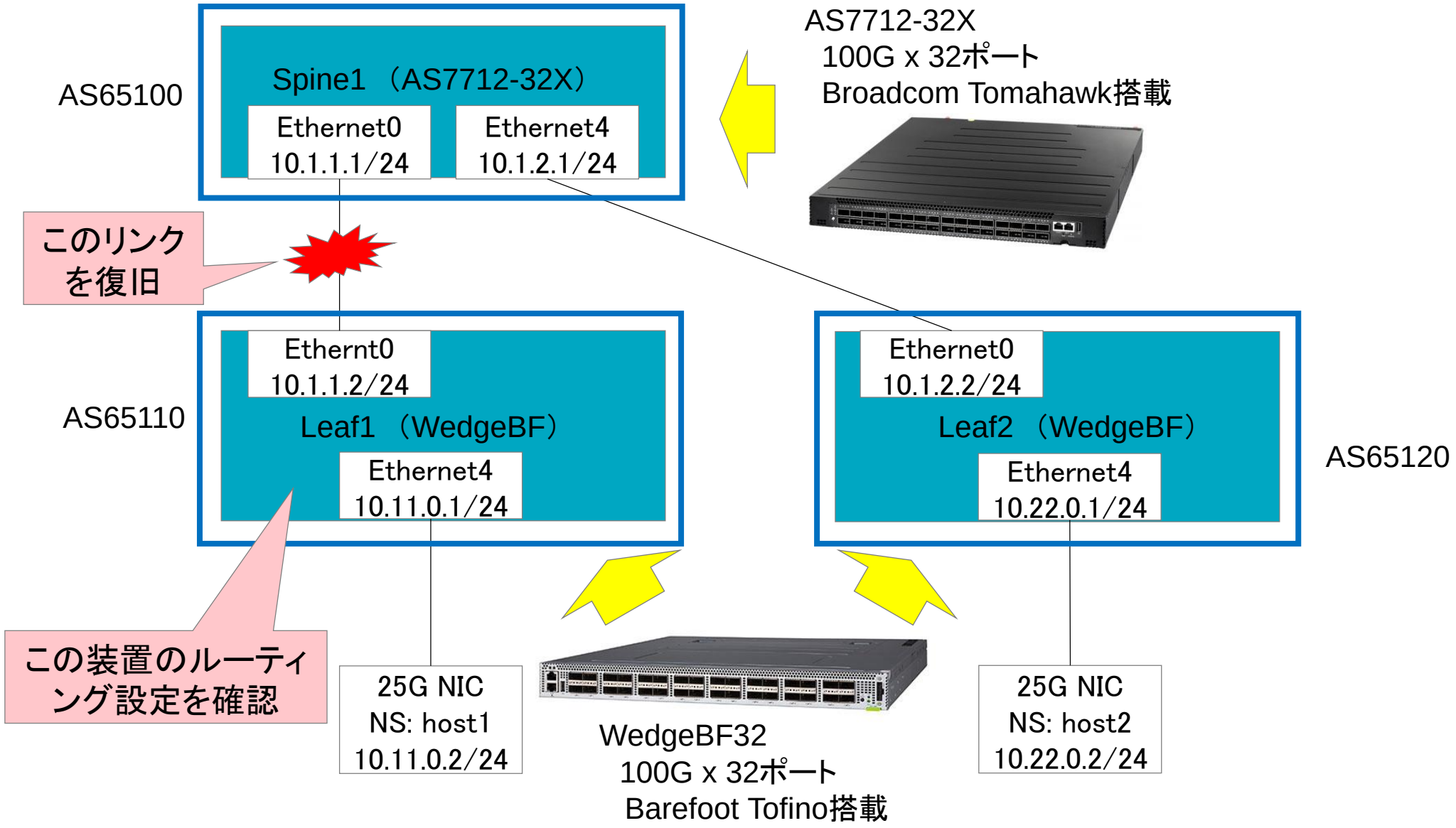
- Quagga伝統のFIB push interfaceを使用
  - http://docs.frrouting.org/en/latest/zebra.html#zebra-fib-push-interface
  - FIBやNetlinkを参照する方法もあるが、OS等の環境依存あり

Routing-state interactions
https://github.com/Azure/SONiC/wiki/Architecture





試験構成 Broadcom x 1 + Barefoot x 2

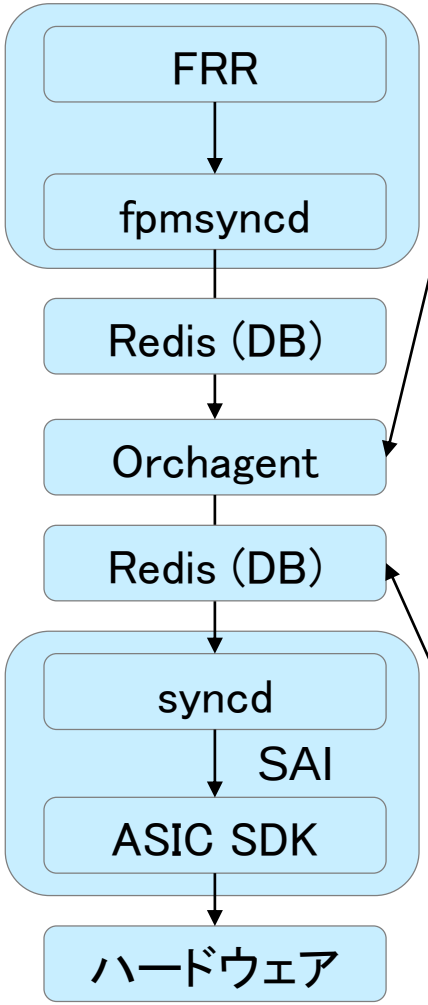


実機にてSONiCを起動

```
admin@Spine:~$ show platform summary
Platform: x86_64-accton_as7712_32x-r0
HwSKU: Accton-AS7712-32X
ASIC: broadcom
admin@Spine:~$ show interfaces status
  Interface            Lanes    Speed     MTU          Alias    Vlan    Oper    Admin    Type    Asym PFC
-----------  ---------------  -------  -------  -------------  ------  ------  -------  ------  ----------
  Ethernet0      49,50,51,52      40G     9100  hundredGigE1  routed      up       up     N/A         N/A
  Ethernet4      53,54,55,56      40G     9100  hundredGigE2  routed      up       up     N/A         N/A
  Ethernet8      57,58,59,60      N/A     9100  hundredGigE3  routed    down       up     N/A         N/A
 Ethernet12      61,62,63,64      N/A     9100  hundredGigE4  routed    down       up     N/A         N/A
...
```

```
admin@Leaf1:~$ show platform summary
Platform: x86_64-accton_wedge100bf_32x-r0
HwSKU: montara
ASIC: barefoot
admin@Leaf1:~$ show interfaces status
  Interface            Lanes    Speed     MTU          Alias    Vlan    Oper    Admin    Type    Asym PFC
-----------  ---------------  -------  -------  -------------  ------  ------  -------  ------  ----------
  Ethernet0          0,1,2,3      40G     9100      Ethernet0  routed      up       up     N/A         N/A
  Ethernet4                4      25G     9100      Ethernet4   trunk      up       up     N/A         N/A
  Ethernet5                5      25G     9100      Ethernet5  routed      up       up     N/A         N/A
  Ethernet6                6      25G     9100      Ethernet6  routed    down       up     N/A         N/A
  Ethernet7                7      25G     9100      Ethernet7  routed    down       up     N/A         N/A
  Ethernet8          8,9,10,11     100G     9100      Ethernet8  routed    down       up     N/A         N/A
...
```

リンクアップ時のルーティング更新のログ




```
ROUTE_TABLE:10.1.1.0/24|SET|nexthop:|ifname:Ethernet0
ROUTE_TABLE:10.22.0.0/24|SET|nexthop:10.1.1.1|ifname:Ethernet0
ROUTE_TABLE:1.1.1.1|SET|nexthop:10.1.1.1|ifname:Ethernet0
ROUTE_TABLE:4.4.4.4|SET|nexthop:10.1.1.1|ifname:Ethernet0
ROUTE_TABLE:10.1.2.0/24|SET|nexthop:10.1.1.1|ifname:Ethernet0
```

```
port_state_change|[{"port_id":"oid:0x1000000000002","port_state":"SAI_PORT_OPER_STATUS_UP"}]|
SAI_OBJECT_TYPE_HOSTIF:oid:0xd000000000314|SAI_HOSTIF_ATTR_OPER_STATUS=true
SAI_OBJECT_TYPE_ROUTE_ENTRY:{"dest":"1.1.1.1/32","switch_id":"oid:0x2100000000000","vr":"oid:0x3000000000023"}|SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID=oid:0x4000000000375
SAI_OBJECT_TYPE_ROUTE_ENTRY:{"dest":"10.1.2.0/24","switch_id":"oid:0x2100000000000","vr":"oid:0x3000000000023"}|SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID=oid:0x4000000000375
SAI_OBJECT_TYPE_ROUTE_ENTRY:{"dest":"10.22.0.0/24","switch_id":"oid:0x2100000000000","vr":"oid:0x3000000000023"}|SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID=oid:0x4000000000375
SAI_OBJECT_TYPE_ROUTE_ENTRY:{"dest":"4.4.4.4/32","switch_id":"oid:0x2100000000000","vr":"oid:0x3000000000023"}|SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID=oid:0x4000000000375
```

リンクアップ前後のLinuxのルーティングテーブル

■リンクアップ前

```
admin@Leaf1:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route

C>* 3.3.3.3/32 is directly connected, lo, 01w3d04h
C>* 10.11.0.0/24 is directly connected, Vlan1001, 01w1d00h
C>* 10.249.28.0/23 is directly connected, eth0, 01w3d04h
```

■リンクアップ後

```
admin@Leaf1:~$ show ip route
（中略）
B>* 1.1.1.1/32 [20/0] via 10.1.1.1, Ethernet0, 00:01:07
C>* 3.3.3.3/32 is directly connected, lo, 01w3d04h
B>* 4.4.4.4/32 [20/0] via 10.1.1.1, Ethernet0, 00:01:07
C>* 10.1.1.0/24 is directly connected, Ethernet0, 00:01:08
B>* 10.1.2.0/24 [20/0] via 10.1.1.1, Ethernet0, 00:01:07
C>* 10.11.0.0/24 is directly connected, Vlan1001, 01w1d00h
B>* 10.22.0.0/24 [20/0] via 10.1.1.1, Ethernet0, 00:01:07
C>* 10.249.28.0/23 is directly connected, eth0, 01w3d04h
```

実機のルーティングの
更新はモニタ投影にて

サマリ

- プレゼン内容
  - SONiCの紹介
  - デモ①　KVM環境でSONiC仮想マシンによるIP CLOSファブリック + Telemetry
  - デモ②　実機SONiC (AS7712とWedgeBF32)によるIP CLOSファブリック

- 議論いただきたいこと
  - White boxスイッチを運用することの問題・課題は何でしょうか？
  - NOSの観点でも、ハードウエアの観点でも、ご意見お願いします！
  - 逆に疑問点ありましたら、ぶつけていただいてもOKです！