匠が斬る!OpenStackストレージとネットワークの活用術

net one

ネットワンシステムズの紹介

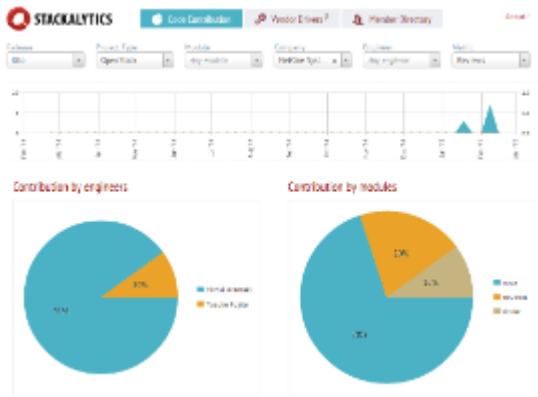
設立：1988年2月1日
社員数：連結2374人(2015年3月31日時点)
連結売上高：1431億(2015年3月期)
本社：千代田区丸の内2-7-2 JPタワー

- 日本における最大手のネットワークインテグレータの一つ
- Ciscoの日本で最初の代理店
- 19インチラックを300台所有する検証テクニカルセンター
- 国内に12の事業拠点
- シリコンバレーとシンガポールに海外オフィス

ネットワンのOpenStackへの取り組み

ネットワンは様々なOpenStackの活動を行っています。

- OpenStack Summit Tokyo / OpenStack Daysへの出展と講演
- 各ベンダーと共同でOpenStackに関連したWhite Paperの作成
- 100名程度のRHEL OpenStack Platform技術者の育成
  (RedHat ForumにてSEアワード受賞)



ネットワンOpenStack Solution

ネットワンはOpenStackのディストリビューションから、各種OpenStackに接続されるインフラや、管理ソフトを提供しています

クラウドコントローラー	CliQr	NetOne Cloud Manager		
OpenStack ディストリビューション	RedHat OpenStack Platform	Ubuntu OpenStack	Cisco Metapod	
OpenStack用 ストレージ	EMC ScaleIO	RedHat Ceph	NetApp	
OpenStack用 ネットワーク	Cisco ACI	BigSwitch	6wind	

NetOne Cloud Manager

ネットワンがOpenStack用に開発したセルフサービスポータルです。安易な操作でアプリケーションや、サーバー、ネットワークの構築が可能です。

NetOne Cloud Manager

仮想ネットワークを構成するためのカタログサービス

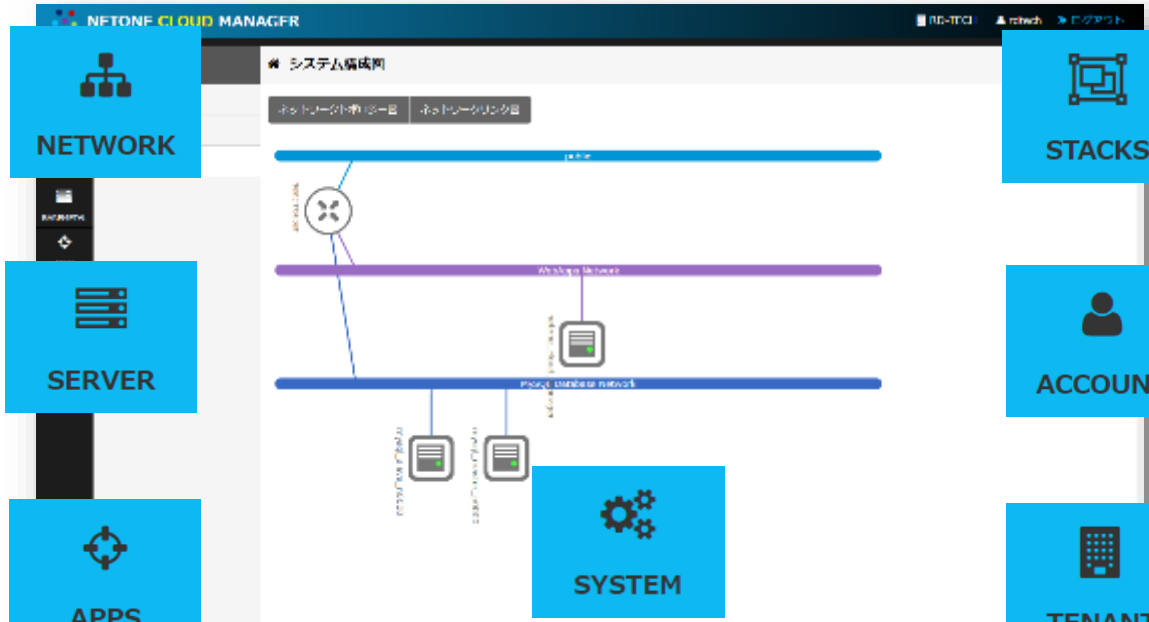
NETWORK

仮想マシンを構成するためのカタログサービス

SERVER

アプリケーションをインストールするためのカタログサービス

APPS



システム管理。新規テナントの発行、コントローラ登録、システムログ管理

STACKS

仮想ネットワーク/マシンをグループで構成するカタログサービス

ACCOUNT

ログインアカウントのプロフィール設定、パスワード変更機能

TENANT

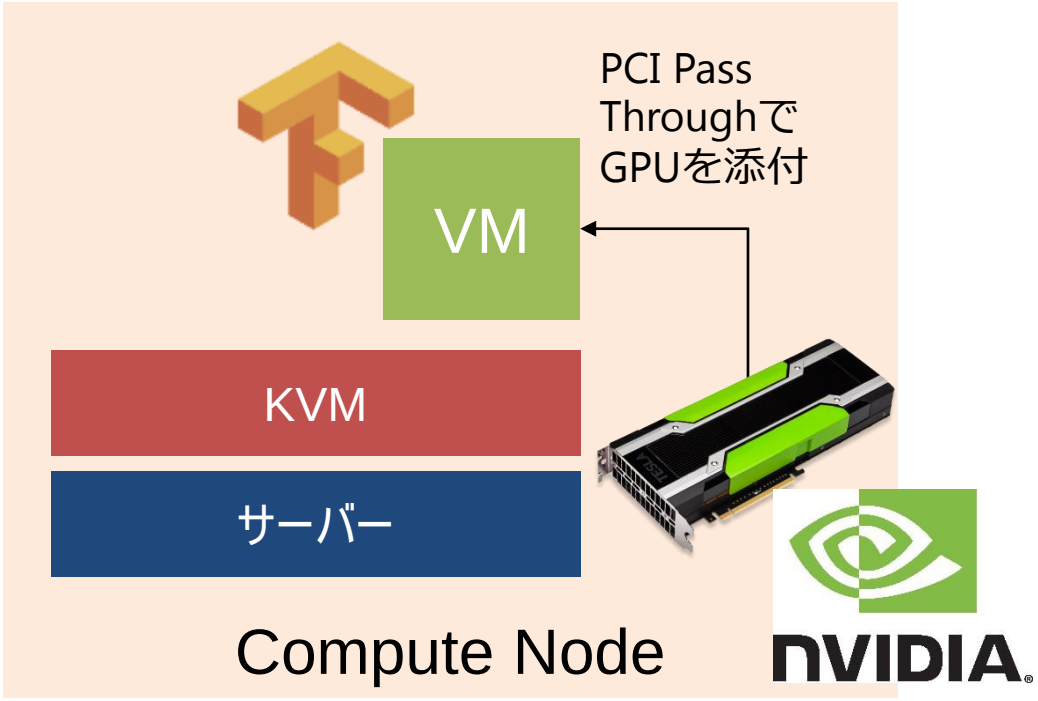
テナント管理。カタログ作成、ユーザ管理、クウォータ利用状況、ログイン履歴管理

OpenStackとDeep Learning

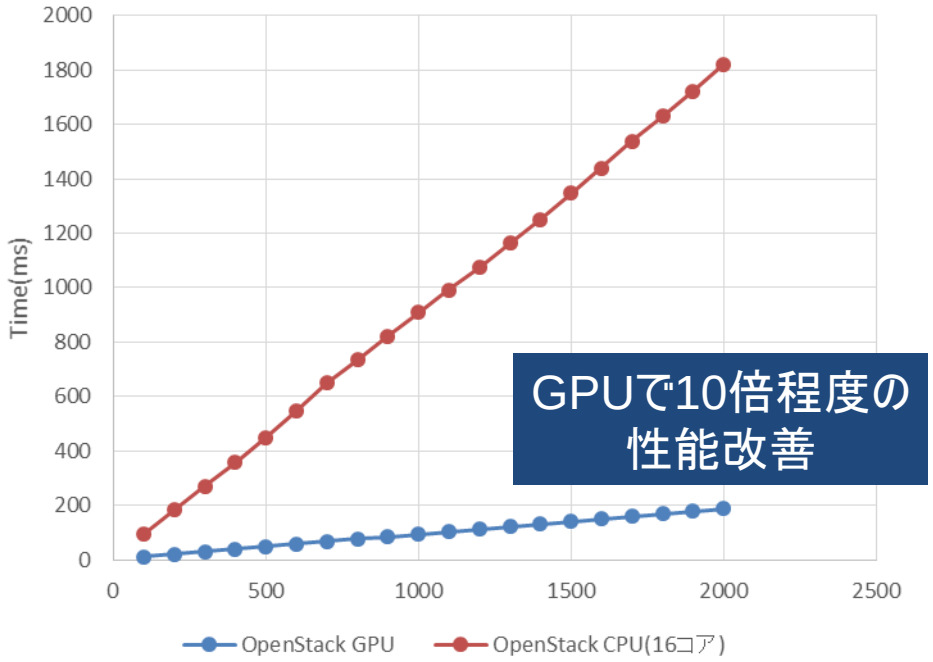
OpenStackを利用して、Deep Learning用の基盤を作成

- OpenStackでGPU付き仮想マシンの作成、削除を簡単に実施
- GPU付きで高速に計算
- 分散計算のノードも簡単作成

- RedHat OpenStack Platform
- Cisco UCS M240
- NVIDIA Tesla M60



Tensorflowを使って、
CNNで学習の計算時間を計測

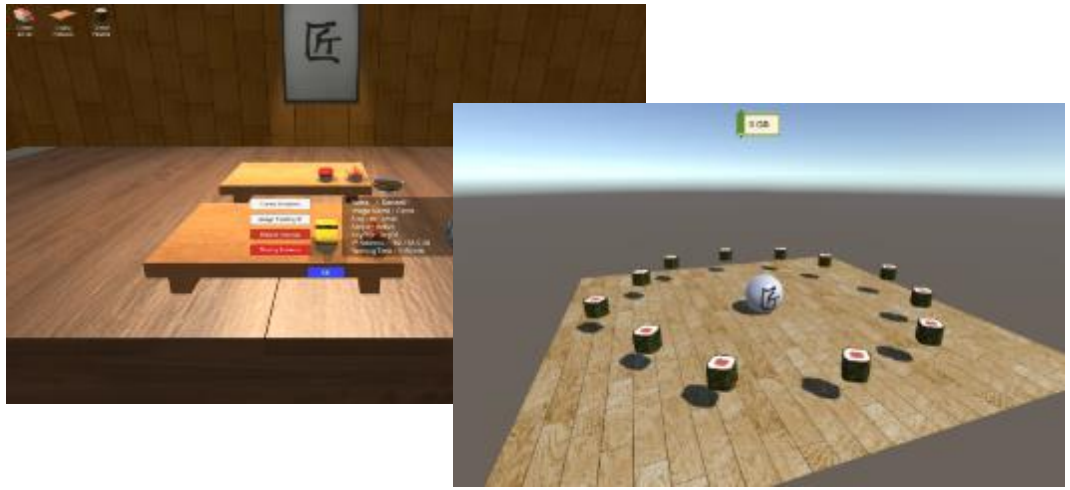


GPUで10倍程度の
性能改善

OpenStackのUnityダッシュボード

OpenStack APIの利用例として、ゲーム開発プラットフォームであるUnityを利用して、
OpenStackのダッシュボードを開発

Unity OpenStack Dashboard
(Sushi Stack)



Virtual Reality OpenStack Dashboard
(Sushi VR)

OpenStackと3rd Party連携

OpenStackが選ばれる理由

オープンソース

オープンな技術で開発されているため、ベンダーロックインを避けることが出来る

豊富な3rd Party連携

ITベンダ各社が標準的に、OpenStack連携のモジュールを提供

API Firstによる開発

APIから実装されるため、他システムとの連携が容易にできる

活発なコミュニティ

半年毎の新規リリースで、数百社、数千人が開発に関与

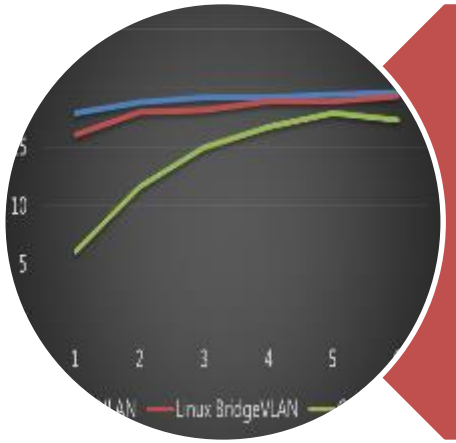
OpenStackで3rd Party連携を行う理由

OpenStackでは機能の向上、性能の改善、既存機器の活用のために3rd Party連携を行います

機能	• 標準では無い機能が必要 • 可用性を高めたい • 可視化したい
性能	• 専用機器による速度や性能が欲しい • 特殊なソフトウェアによる性能向上 • ハードウェアによる速度や、オフロード機能の利用
既存の活用	• 新たな投資を最小限にしたいため、既存の機器を利用 • 既存の仮想化環境、ストレージ、ネットワークを利用

3rd Party連携へのネットワンのアプローチ

標準的な実装と、3rd Party連携した場合の差異と良い点がわからない



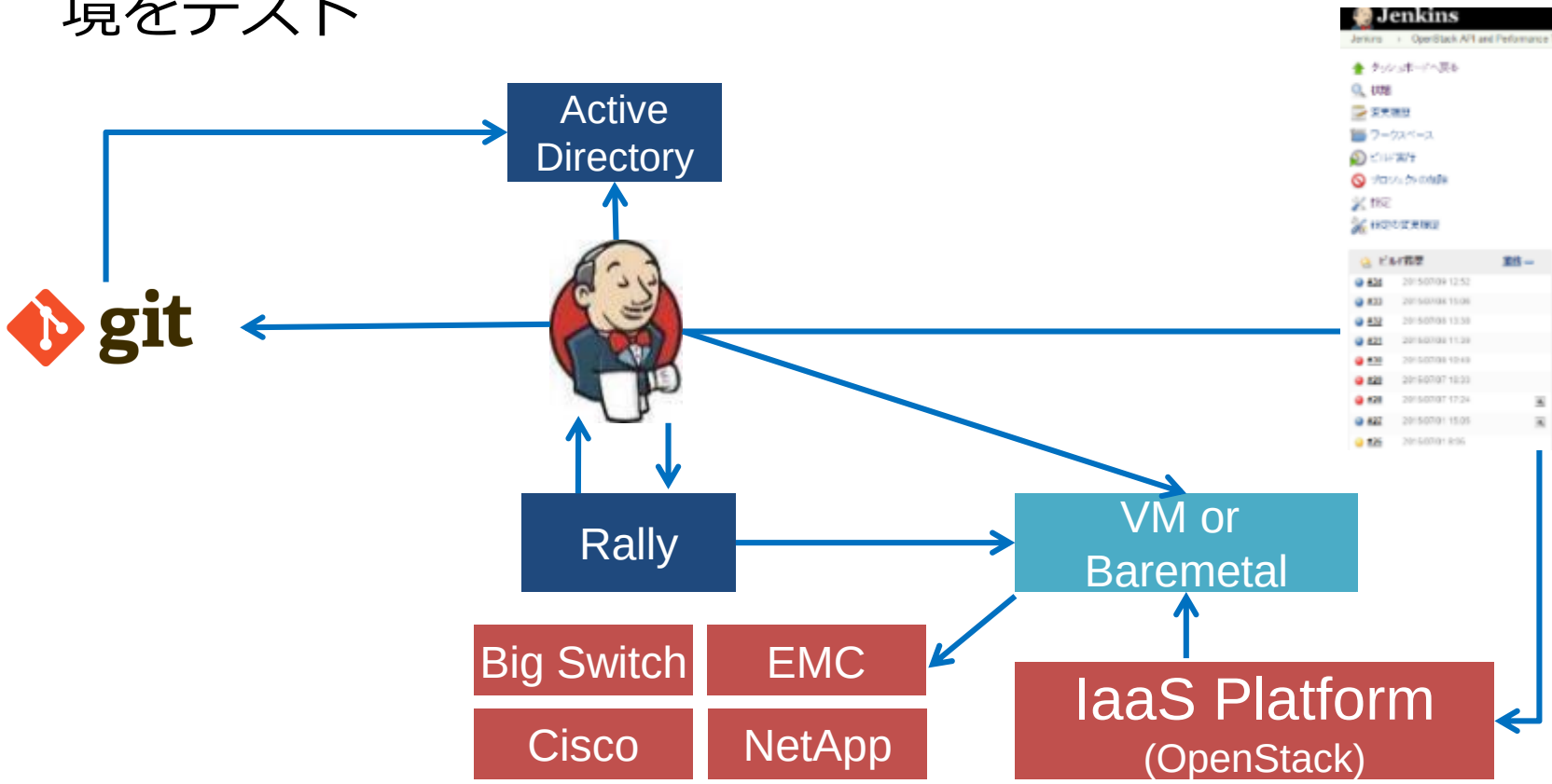
デフォルトを理解して3rd Party連携の強みを確認

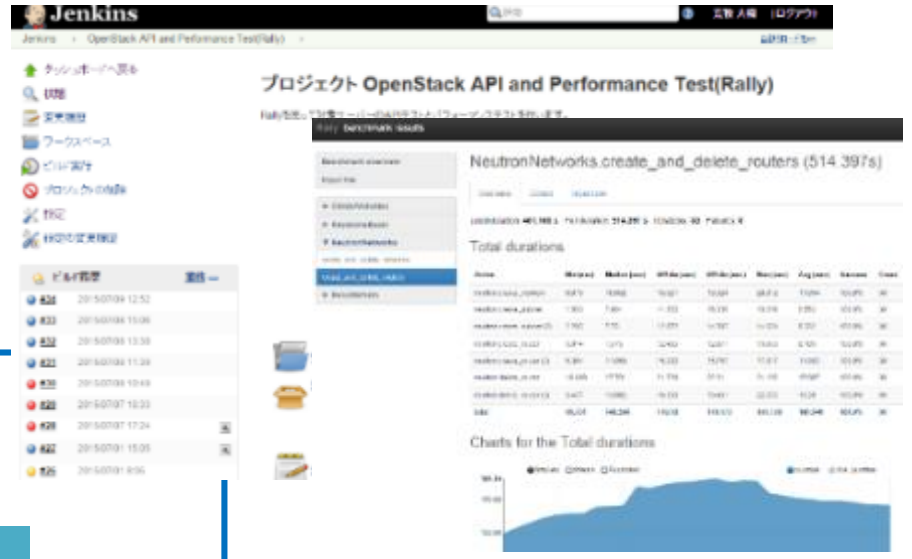
OpenStack、3rd Partyのソフトウェアがそれぞれ Updateされているため手動でのIntegrationテストが現実的で無い

テストを自動化

3rd Party連携テストの自動化

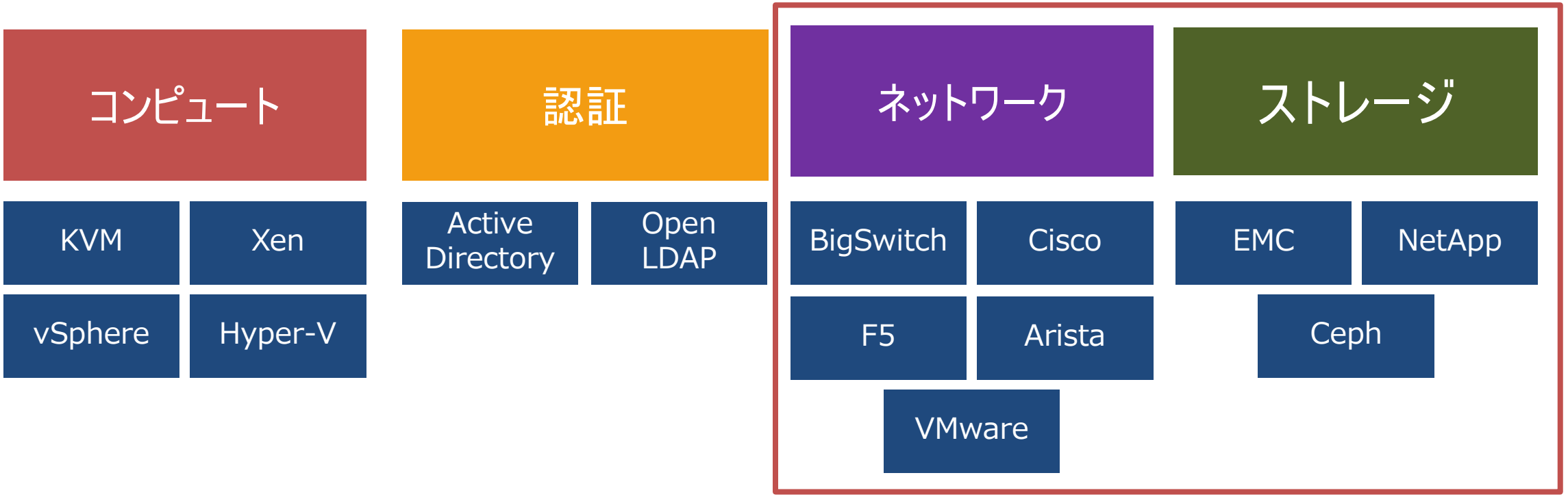
- Jenkinsを利用して、3rd Partyテストを自動化
- OpenStackのプロジェクトであるRallyを使って3rd Party連携後のOpenStack環境をテスト





OpenStackと3rd Party連携について

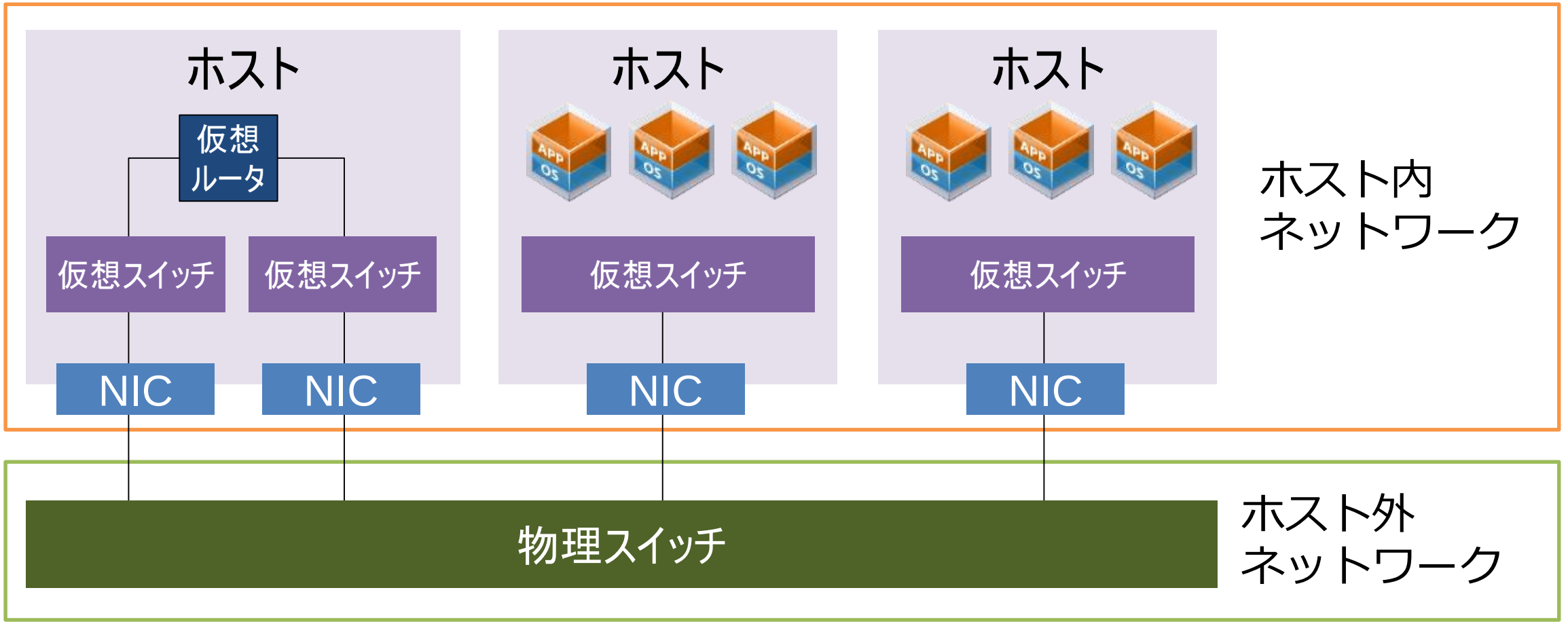
OpenStackは様々な部分で、3rd Party連携が可能ですが、今回はネットワークとストレージの連携にフォーカスを当てます



OpenStackとネットワークについて

OpenStackネットワークの考慮点

OpenStackではホスト内のNWと、ホストの外のNWと両方を考える必要があります。



ネットワンのOpenStack NWへの取り組み

ホスト内の仮想SWと、物理スイッチ連携と2つのテーマで取り組みを行っています

ホスト内仮想スイッチ

Open vSwitch

Linux Bridge

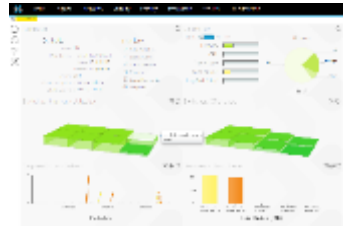
WIND

DPDK
DATA PLANE DEVELOPMENT KIT

物理スイッチ連携

Cisco

big switch networks

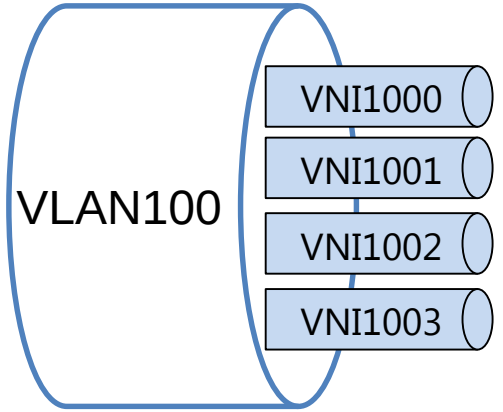


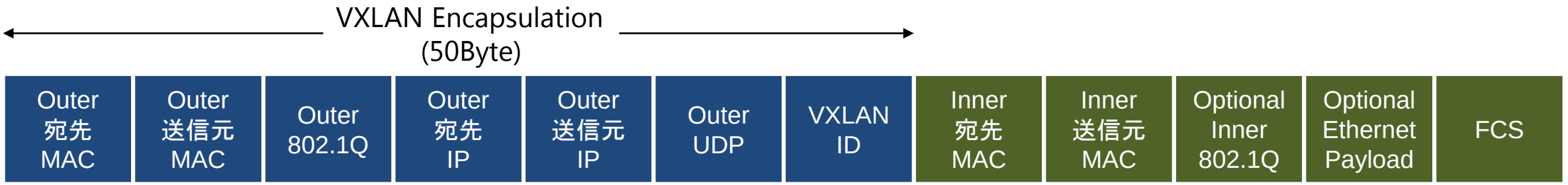
ホスト内ネットワーク

OpenStackとネットワークとホスト内NW

OpenStackは様々な方式を選択可能ですが、標準的にはOpen vSwitchとVXLANを組み合わせた構成が取られます。VXLANはVLANを拡張して、1600万個のNWの作成が可能となります。

	VLAN	VXLAN
NW数	4096	約1600万
MTU	1500 bytes	1550 bytes 以上 (物理スイッチ側で対応必要)
NW情報	通常のヘッダ情報の中に含まれる	VXLANヘッダが追加されて、VXLANの情報が格納される





様々なホスト内ネットワークの選択肢

ホスト内のネットワーク構成は様々な選択肢があります。標準のOvSより機能と性能の向上を目指す場合に様々な選択肢があります。

	OVS	OVS-DPDK	6 WIND	SR-IOV
セグメンテーション	Flat/VLAN/VXLAN/その他	Flat/VLAN	Flat/VLAN/VXLAN	Flat/VLAN
セキュリティグループ	○	x	○	x
高スループット時のCPU使用率	高い	低い	低い	低い
速度	△	◎ (機能が単純なため速度が出やすい)	○	◎
Live Migration	○	○	○	X
概要	OpenStack標準の構成	OVSをDPDKで利用する	6Windの開発DPDKの商用ソフトウェア	物理NICを仮想的に仮想マシンに見せる

OpenStackのホスト内仮想スイッチを利用する場合の疑問

OpenStack内の仮想スイッチを利用する上での様々な疑問点

OVSで十分なパフォーマンスは出るのか?

OVSとLinux Bridgeで性能差はあるのか?

VXLANはパフォーマンスに影響を及ぼすのか?

DPDKを利用する事でパフォーマンスは改善するか?

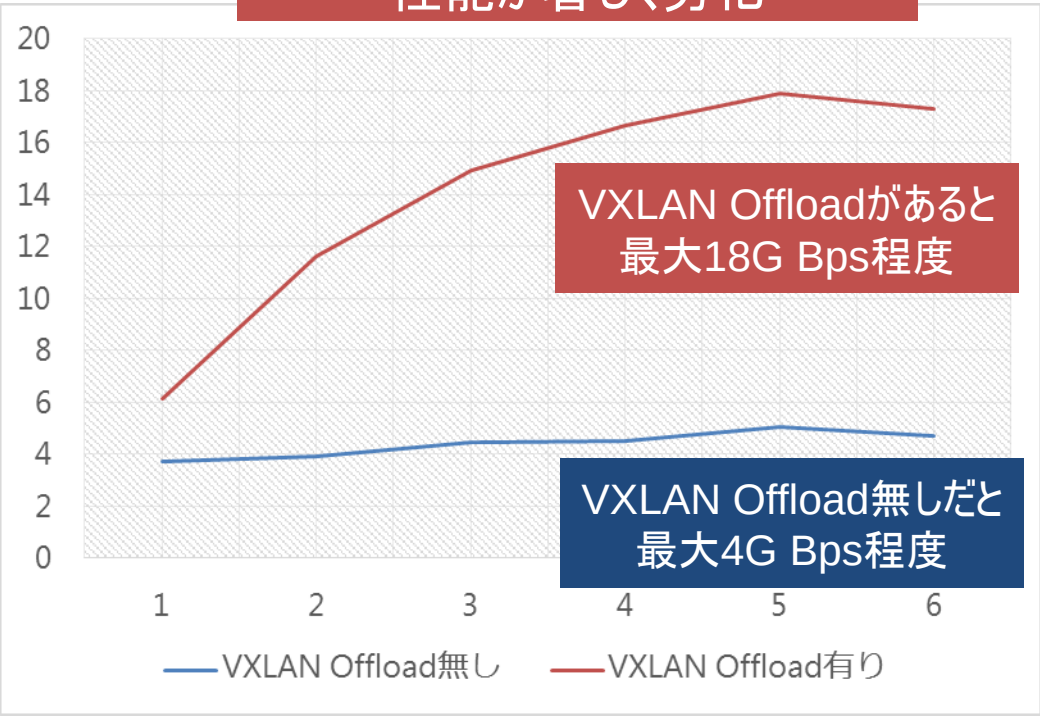
仮想スイッチでRoutingやNATの性能は十分か?

実際に計測して疑問を解消

ネットワークの計測例

計測を行うとDPDKの効果や、VXLAN Offload NICの効果が顕著に出てきます

VXLAN Offload NICを
利用しない場合は
性能が著しく劣化

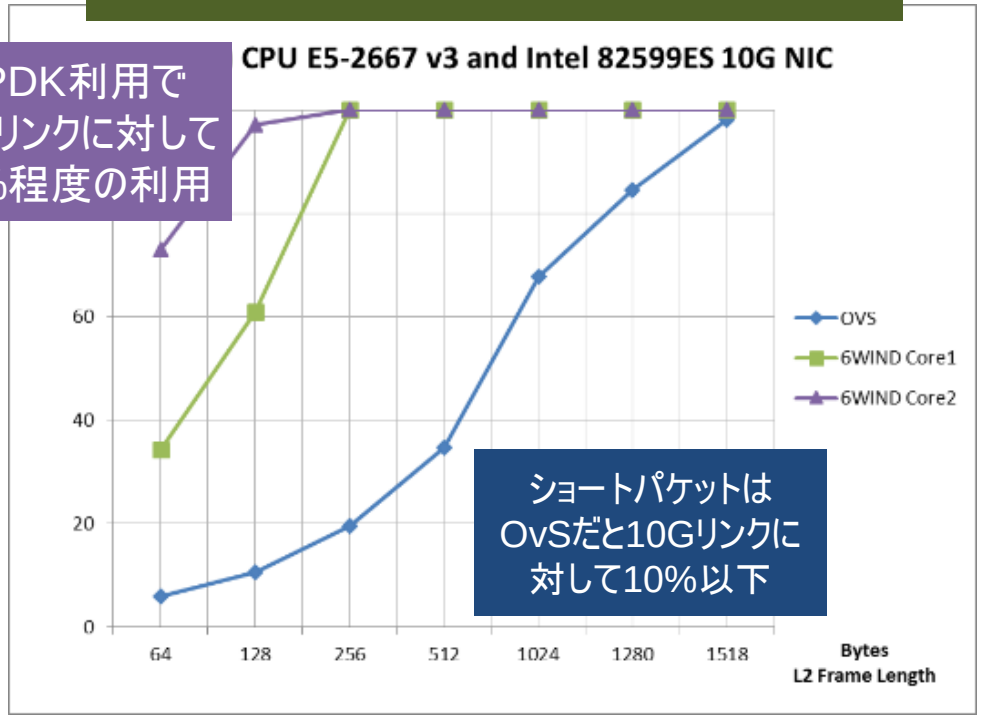


VXLAN Offloadがあると
最大18G Bps程度

VXLAN Offload無しだと
最大4G Bps程度

OvSの苦手とするショートパケットを
DPDKで性能を改善

DPDK利用で
10Gリンクに対して
70%程度の利用



ショートパケットは
OvSだと10Gリンクに
対して10%以下

計測からわかる様々な結果

実際の計測を行うと様々な結果がわかります。

OVSとLinux Bridgeでの
性能差は無い

NATとルータでの性能
劣化は10%-15%程度

OVSはショートパケットで
大きな性能劣化

VXLAN Offload NICを
利用しない場合は
性能が著しく劣化

DPDKを利用すれば
大幅な性能改善
ショートパケットの性能改善は顕著

実際に計測すると様々面が見えてくるので、疑問点の
計測は重要

ホスト内仮想スイッチについて

パフォーマンス	• NIC辺り10G出れば十分。ショートパケットの考慮はしない。 → OVSを利用 • ショートパケット含めたパフォーマンスが必要 → DPDKを利用した製品を検討(OVS-DPDK/6wind)
機能	• NW機能が重要(VXLANやセキュリティグループ) → 6wind / OVSを利用 • NW機能は利用しないが、パフォーマンスが必要 → OVS-DPDK
NICの選択	• VXLANを利用 → VXLAN Offload機能付きのNIC • VLANを利用 → 通常のNICのVLAN Offload機能で十分

仮想スイッチのパフォーマンス測定の注意点

1G以上の通信を実施しようとなると様々な状況の影響を受けますので注意が必要です。

NICファームのバージョン	NICの設定	VXLAN Offloadの対応OS
• NICのファームのVersionがスループットに影響を及ぼす • 最新のNICのファームにすると状況が改善	• NICの設定がスループットに影響を及ぼす • NICのQueueの数を増やす事で状況が改善	• RHEL等のLinux環境でVXLAN Offloadに対応しているか注意 • VMware環境のみVXLAN Offload対応しているNICもあり

物理NWとの連携

OpenStackと物理NWの連携

OpenStackと各種SDN製品との深い連携が可能になってきました。

Phase 0	Phase 1	Phase 2	Phase 3
OverlayでNW機器との連携は無し	仮想マシンの配置に応じて物理スイッチ上のVLANを開け閉めする	ML2を利用して、仮想スイッチと、物理スイッチが連携。L3機能を提供	仮想スイッチと、物理スイッチがSDN Controllerから操作される。NATやProxyの機能に完全対応
Open Stack → 仮想スイッチ 連携なし	Open Stack VLANの開け閉めを指示	Open Stack → 仮想スイッチ	Open Stack → SDN Controller 仮想スイッチ
			

Cisco ACIとBigswitchのP+V連携

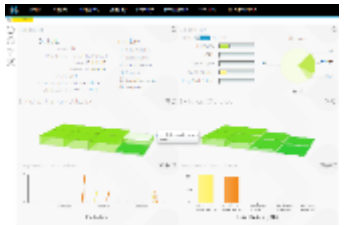
CiscoとBigSwitchで2015年後半から、P+V構成のサポートを始めました

Cisco ACIの特徴

- Group Policyを利用したPolicy BaseのNWへ拡張可能
- Nexus 9kを利用した構成
- P+Vによる物理と仮想スイッチの管理と可視化

Cisco

big switch networks



BigSwitchの特徴

- ホワイトボックスSwitchが利用可能
- GUIによるファブリックと仮想スイッチの管理
- P+Vによる物理と仮想スイッチの管理と可視化

OpenStack- BigSwitch BCF連携

統合

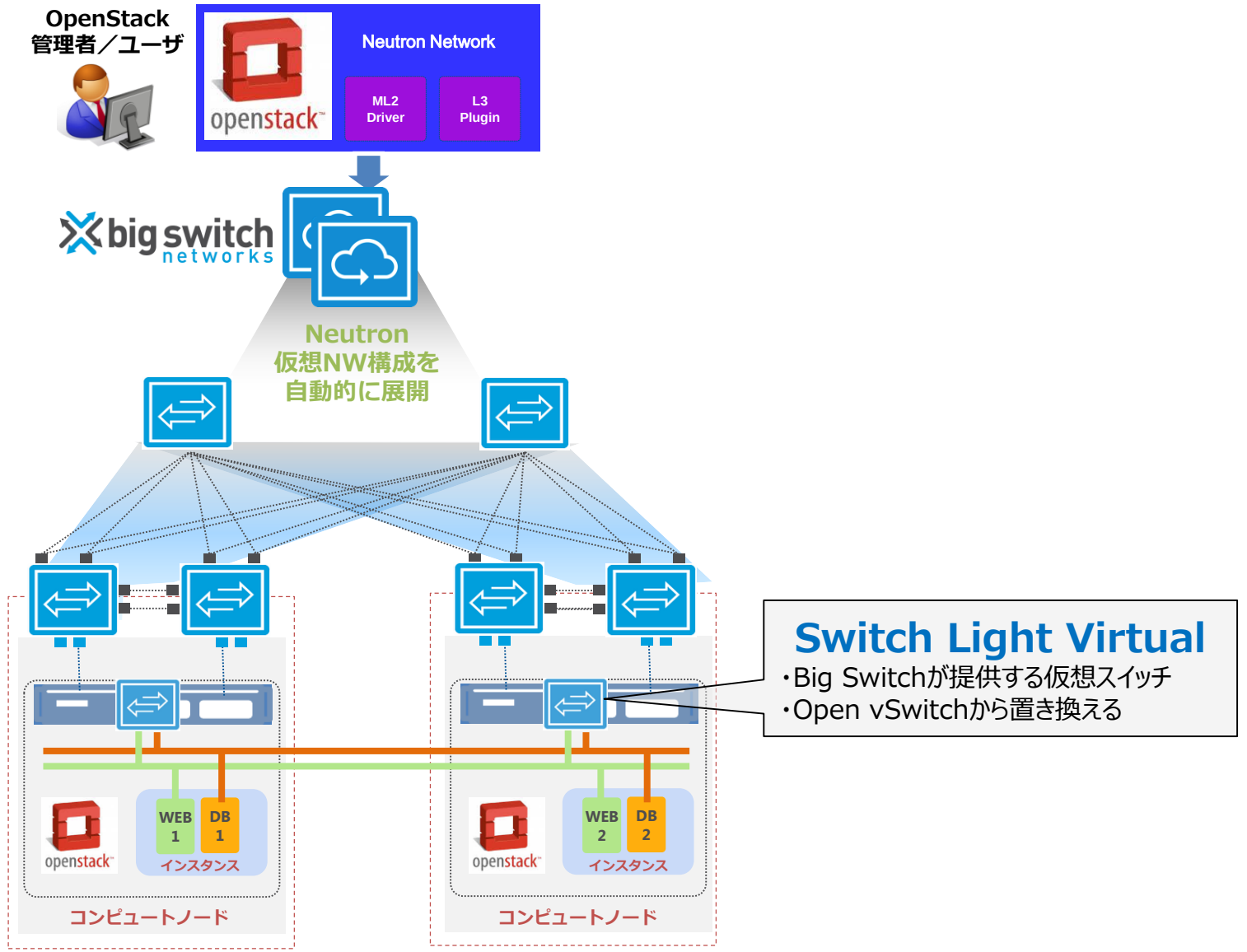
- 物理スイッチと仮想スイッチを統合的に制御・管理
- OpenStackテナントをファブリック上に自動的に展開

分散L3

- 仮想/物理の全スイッチでL3処理
- Neutronの性能ボトルネックを解消

可視化

- BCFコントローラが物理/仮想の全スイッチから情報を収集



Open vSwitchとBig Cloud Fabric(P+V)機能比較

機能	Open vSwitch	Big Cloud Fabric(P+V)
仮想/物理スイッチの統合管理	×	○ 物理トポロジ、論理トポロジ、 HAを統合管理
分散ルーティング(Node間、Node内)	× Network Node処理※	○ 各Compute Nodeで処理
分散ルーティング(Floating IP)	× Network Node処理※	○ 各Compute Nodeで処理
分散ハードウェアフォワーディング	× HSRP/VRRPによるL3ハンドリング	○ プロトコルレス分散フォワーディング
稼働状況や統計の可視化	×	○ 物理/仮想NWの可視化、ログ検索・可視化
Compute Node増設時の自動検知	×	○ NIC Bondingを検知し、Switch側でポートグループを自動構成

※分散ルーティング機能はRed Hat OpenStack PlatformのOpen vSwitchではTech Previewのため

OpenStack-BCF連携を利用する場合の疑問

BCFを利用する上での様々な疑問点

BCFで本当に物理/仮想NW可視化ができるのか

OVSとBCFで性能差はあるのか?

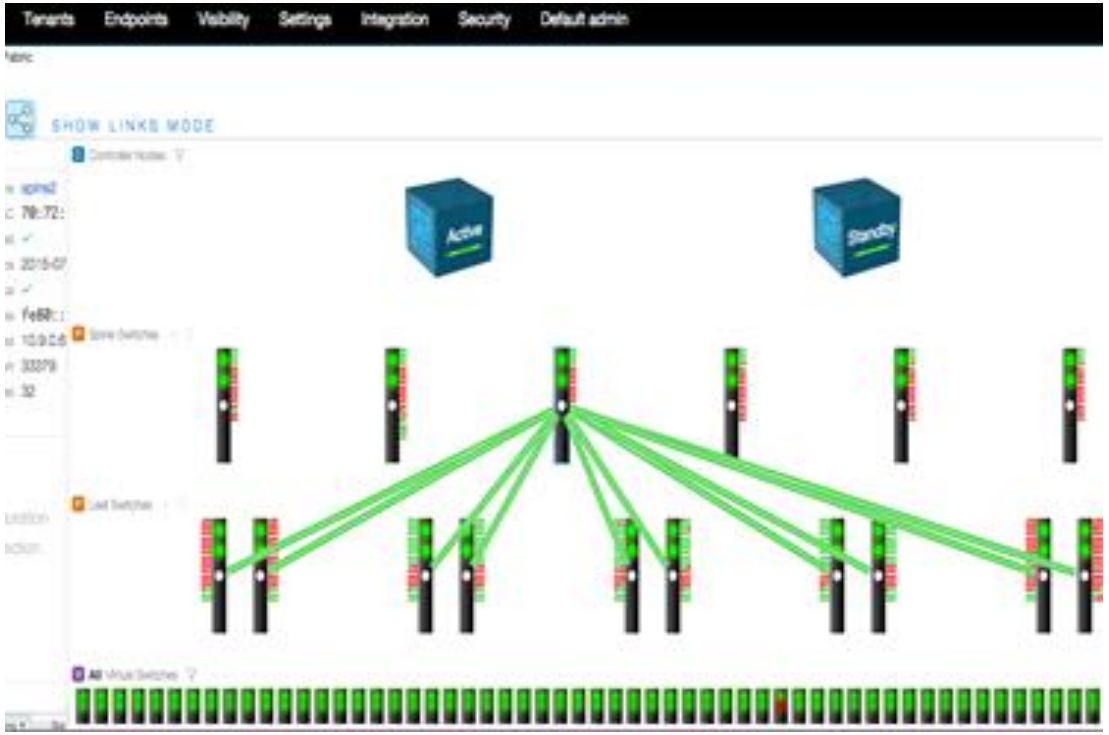
BCFを用いることによってボトルネックは解消されるのか?

BCFはテナントNWに対応しているのか?

BCFを導入する際の注意点は何か?

実際に評価して疑問を解消

物理/仮想ネットワークの可視化



Diagnostic | Properties

Interface Name	Status	Admin Status	Link Status	Hardware Address	Number	Port Group	Storm Control P
enp9s0	Link Down	Up	Down	84:b8:02:d2:29:b3	1	–	–
enp129s0f0	Up	Up	Up	00:90:fa:8b:35:5e	2	–	–
inband	Up	Up	Up	6e:72:77:95:9e:2f	1000	–	–
local	Link Down	Up	Down	be:1b:0e:7e:3e:70	4294967294	–	–
nat5c16c7030000	Up	Up	Up	22:78:97:df:7d:de	3	–	–
nat5c16c7040000	Up	Up	Up	3e:8c:49:48:a1:3f	5	–	–
qvo03ed0868-15	Up	Up	Up	ba:2c:89:40:fe:c2	9	–	–
qvo6e3bb2a6-00	Up	Up	Up	42:8c:8f:35:28:d4	11	–	–
qvo86c1acfb-82	Up	Up	Up	66:64:80:bd:1e:79	8	–	–
qvo6138e689-c3	Up	Up	Up	d2:0c:57:a0:0b:70	6	–	–

ファブリックトポロジ
- コントローラ、物理＋仮想スイッチの可視化
- 接続状態の視覚化

仮想スイッチ
- 仮想インタフェース可視化

テナントネットワークの可視化

Tenant	Segment	Name	Attachment State	Description	IP Addresses	MAC Address	Vendor	NAT Endpoint	Switch	Interface
admin.neutron	ext	–	✓ Active	–	10.135.82.253 (learned)	2c:54:2d:bc:d0:bf	Cisco Systems, Inc	–	Leaf3	ethernet48
admin.neutron	ext	–	✓ Active	–	–	b0:fa:eb:6b:f8:20	Cisco Systems, Inc	–	Leaf3	ethernet48
admin.neutron	ext	–	✓ Active	–	10.135.82.154 (static)	5c:16:c7:04:00:00	Big Switch Networks	✓	sbc-rhosp-com01.noslocal.com	nat5c16c7040000
admin.neutron	ext	6138e689-c337-4abb-bb71-dbd975f...	✓ Active	sbc-rhosp-com01.noslocal.com:qvo6138e689-c3	10.135.82.157 (static)	fa:16:3e:c5:4d:a0	–	–	sbc-rhosp-com01.noslocal.com	qvo6138e689-c3
admin.neutron	ext	69927163-0e05-453e-a594-1916a0...	✓ Active	sbc-rhosp-com02.noslocal.com:qvo69927163-0e	10.135.82.158 (static)	fa:16:3e:df:b8:11	–	–	sbc-rhosp-com02.noslocal.com	qvo69927163-0e
admin.neutron	ext	cf1d086f-e439-46c6-a1fc-93b9c041...	✓ Active	sbc-rhosp-com01.noslocal.com:tapcf1d086f-e4	10.135.82.151 (static)	fa:16:3e:49:12:4b	–	–	sbc-rhosp-com01.noslocal.com	tapcf1d086f-e4
project1.neutron	internal	025d9084-8ab1-4fcb-a4e6-743737d...	✓ Active	sbc-rhosp-com02.noslocal.com:qvo025d9084-8a	10.0.0.8 (static)	fa:16:3e:e4:4b:65	–	–	sbc-rhosp-com02.noslocal.com	qvo025d9084-8a
project1.neutron	internal	03ed0868-15a6-4284-8479-31b6f69...	✓ Active	sbc-rhosp-com01.noslocal.com:qvo03ed0868-15	10.0.0.9 (static)	fa:16:3e:1a:94:10	–	–	sbc-rhosp-com01.noslocal.com	qvo03ed0868-15
project1.neutron	internal2	2d567aef-35d9-4138-a47c-124ee50...	✓ Active	sbc-rhosp-com02.noslocal.com:qvo2d567aef-35	20.0.0.3 (static)	fa:16:3e:43:0a:e9	–	–	sbc-rhosp-com02.noslocal.com	qvo2d567aef-35
project1.neutron	internal2	2edb1053-97c7-4f95-964b-c0346de...	✓ Active	sbc-rhosp-com01.noslocal.com:tap2edb1053-97	20.0.0.2 (static)	fa:16:3e:4a:fb:e3	–	–	sbc-rhosp-com01.noslocal.com	tap2edb1053-97
project1.neutron	internal	324ef583-ec6a-4629-b358-f071db6...	✓ Active	sbc-rhosp-com02.noslocal.com:qvo324ef583-ec	10.0.0.6 (static)	fa:16:3e:e4:de:06	–	–	sbc-rhosp-com02.noslocal.com	qvo324ef583-ec
project1.neutron	internal2	58552d97-596e-4a4f-97cf-7dff61be...	✓ Active	sbc-rhosp-com01.noslocal.com:qvo58552d97-59	20.0.0.6 (static)	fa:16:3e:da:d9:d5	–	–	sbc-rhosp-com01.noslocal.com	qvo58552d97-59
project1.neutron	internal2	6e3bb2a6-00e5-4e08-b606-1ac3012..	✓ Active	sbc-rhosp-com01.noslocal.com:qvo6e3bb2a6-00	20.0.0.5 (static)	fa:16:3e:5e:c3:ba	–	–	sbc-rhosp-com01.noslocal.com	qvo6e3bb2a6-00
project1.neutron	internal	86c1acfb-820a-4382-9c1e-31b96a8...	✓ Active	sbc-rhosp-com01.noslocal.com:qvo86c1acfb-82	10.0.0.7 (static)	fa:16:3e:9a:2f:a1	–	–	sbc-rhosp-com01.noslocal.com	qvo86c1acfb-82
project1.neutron	internal2	a1f0b5e0-428d-419c-9609-6600ce8...	✓ Active	sbc-rhosp-com02.noslocal.com:qvoa1f0b5e0-42	20.0.0.4 (static)	fa:16:3e:d0:ae:e2	–	–	sbc-rhosp-com02.noslocal.com	qvoa1f0b5e0-42
project1.neutron	internal	d1cc0961-2d92-435a-8dc6-3cc4022...	✓ Active	sbc-rhosp-com02.noslocal.com:qvod1cc0961-2d	10.0.0.10 (static)	fa:16:3e:d1:fc:75	–	–	sbc-rhosp-com02.noslocal.com	qvod1cc0961-2d
project1.neutron	internal	d567b6f8-923c-408c-8137-7dbd422...	✓ Active	sbc-rhosp-com02.noslocal.com:tapd567b6f8-92	10.0.0.2 (static)	fa:16:3e:5b:8a:cc	–	–	sbc-rhosp-com02.noslocal.com	tapd567b6f8-92
project1.neutron	internal	ebb35e53-454c-4035-a3e4-c189ec4...	✓ Active	sbc-rhosp-com01.noslocal.com:qvoebb35e53-45	10.0.0.4 (static)	fa:16:3e:78:00:80	–	–	sbc-rhosp-com01.noslocal.com	qvoebb35e53-45
project1.neutron	internal2	ef1a136e-e59b-41b4-8018-917bea...	✓ Active	sbc-rhosp-com02.noslocal.com:qvoef1a136e-e5	20.0.0.7 (static)	fa:16:3e:ce:e9:11	–	–	sbc-rhosp-com02.noslocal.com	qvoef1a136e-e5

OpenStackテナントのエンドポイント

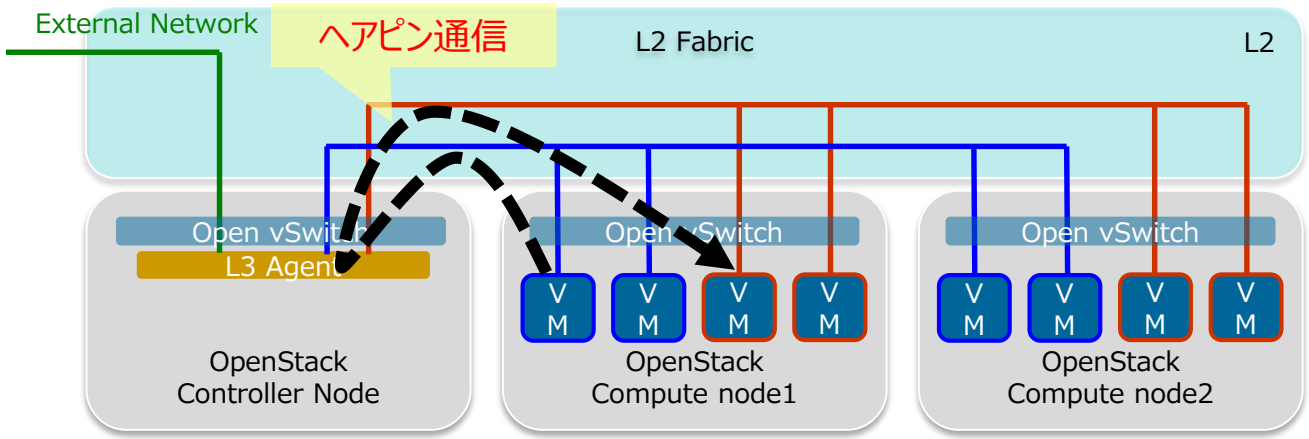
- テナントネットワークとその状態
- 仮想インタフェースのIPアドレスとMACアドレス
- 仮想インタフェースの接続先

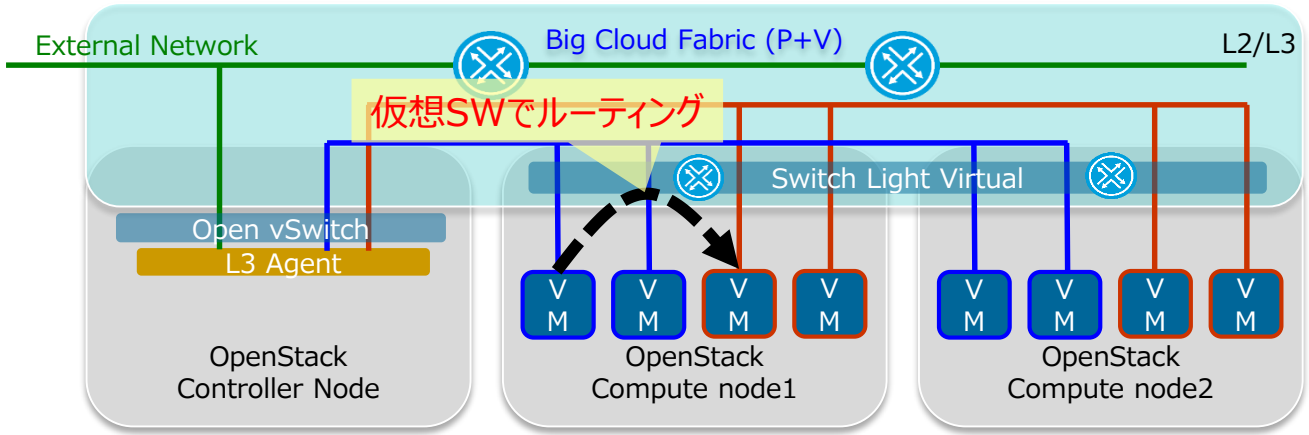
Open vSwitchとBig Cloud Fabric(P+V)の性能比較

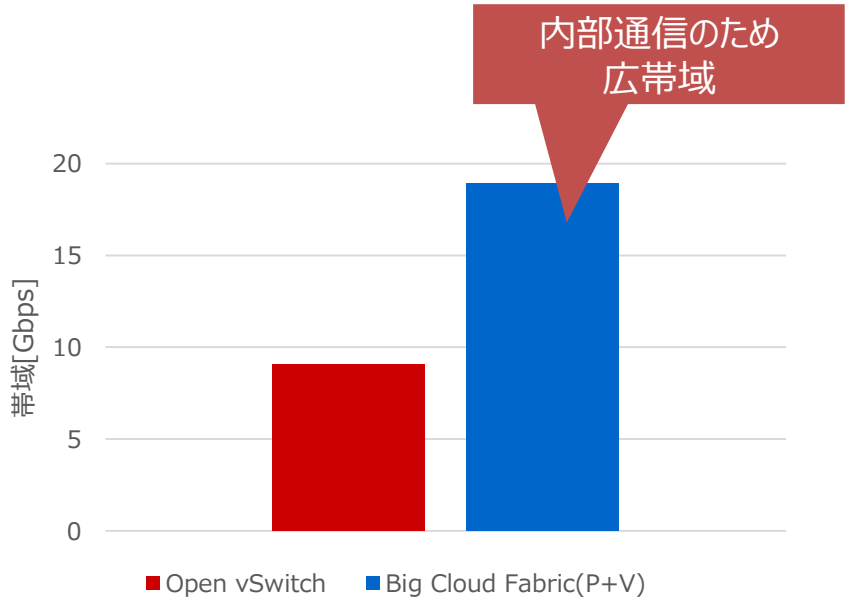
Open vSwitch　vs　Big Cloud Fabric(P+V)

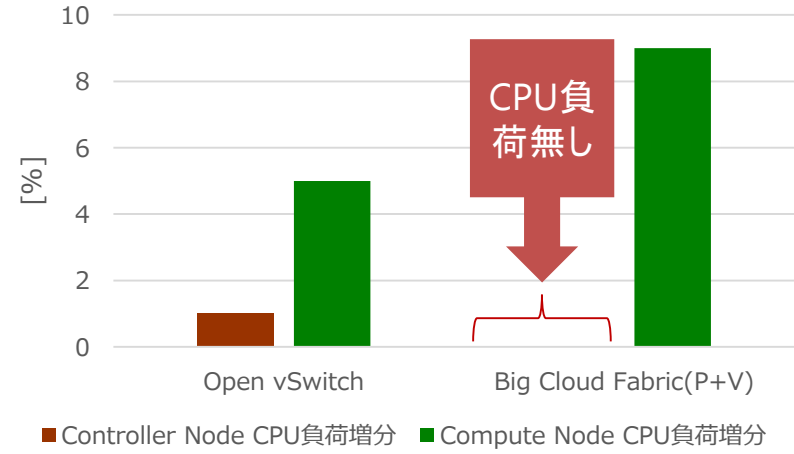
同一Compute Node内の仮想インスタンスのL3通信

- BCFの通信は内部で閉じるため広帯域
- BCFは内部通信のため、Controller Node側のCPU負荷が少ない





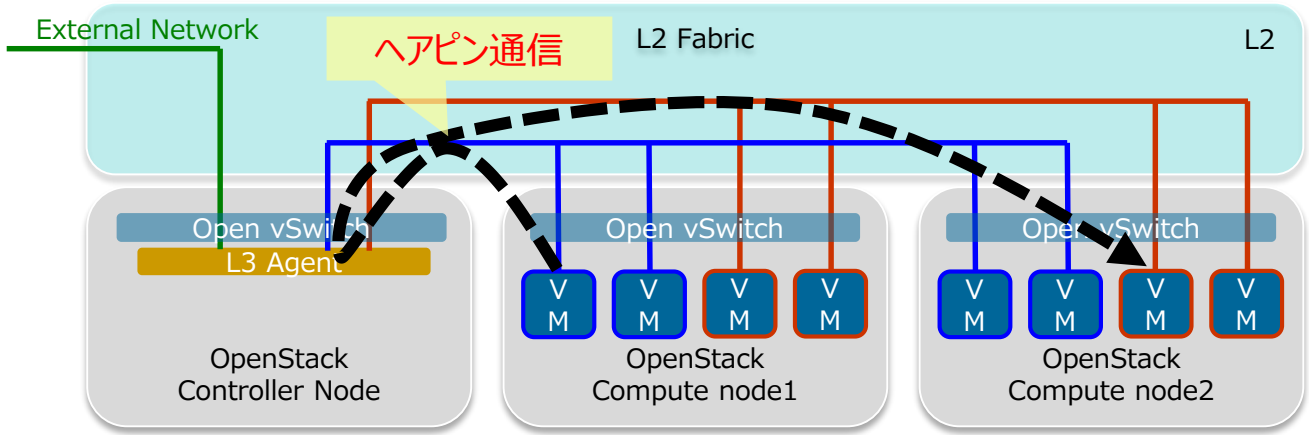


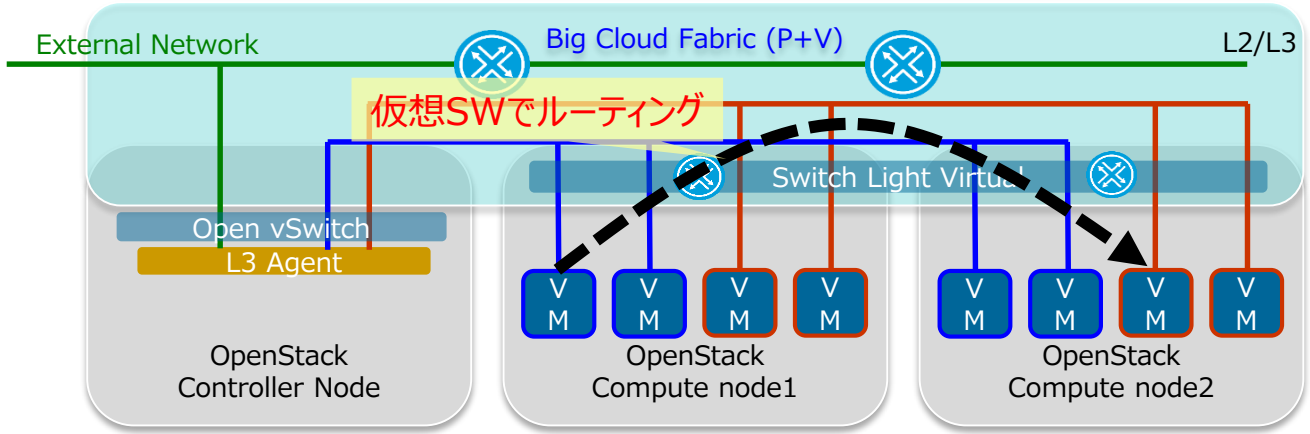


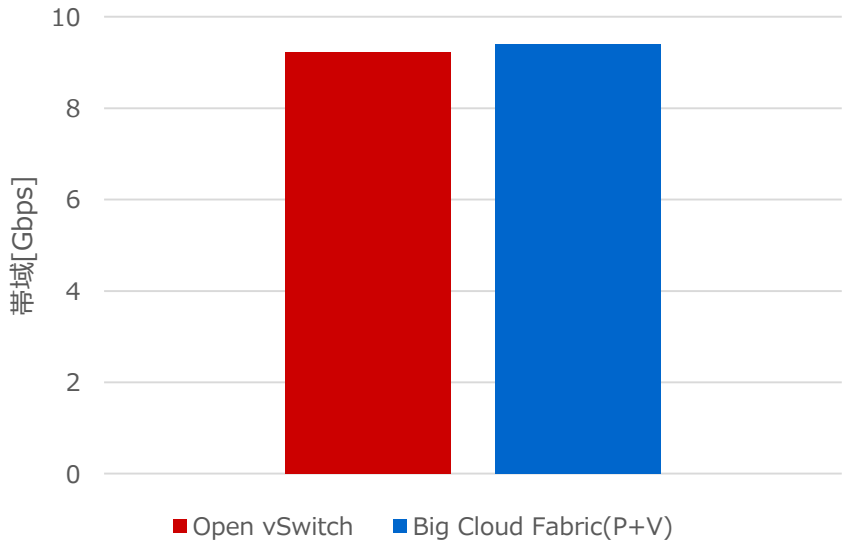
Open vSwitch　vs　Big Cloud Fabric(P+V)

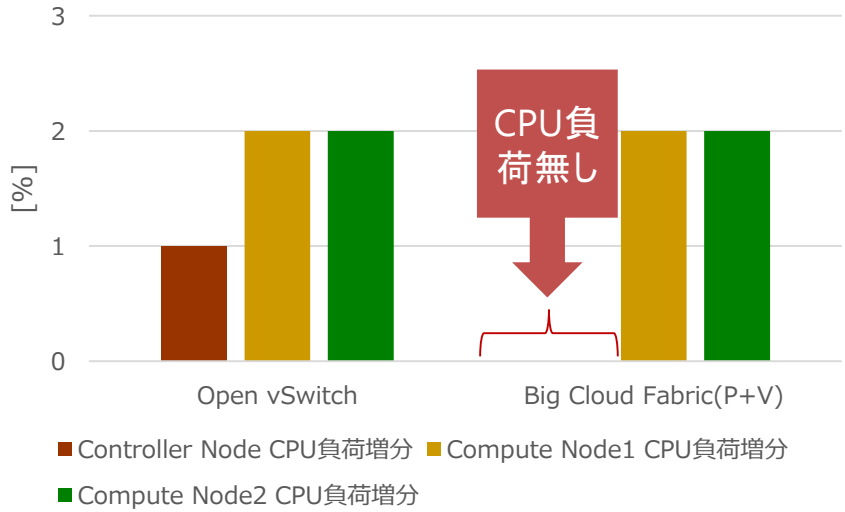
異なるCompute Node間の仮想インスタンスのL3通信

- BCFは仮想スイッチでルーティングされるため、Controller Node側のCPU負荷が少ない
- 仮想スイッチの負荷も殆ど見られない









検証からわかる様々な結果

実際に検証すると様々な結果がわかります。

仮想と物理を可視化可能

BCFでL3内部通信により性能向上

BCF(P+V)はNetwork Nodeを経由せずボトルネック回避

BCFはテナントのエンドポイントまで把握

BCF(P+V)はVLANタグなしLLDP対応NICが必要

実際に評価して疑問を解消

検証の結果とWhite Paperについて

検証の詳しい結果はBig Switch Networks、
レッドハットと共同執筆したWhite Paperに
記述致しました

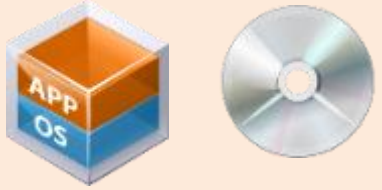
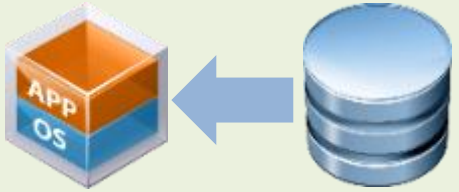
ネットワンブースにお越しください。

本セッションの内容を盛り込んだ
ホワイトペーパーを差し上げます。

ストレージとの連携

OpenStackとストレージについて

OpenStackとストレージは4つの利用方法があります

OpenStack 基盤でストレージを利用	OpenStackサービスでストレージを利用		
	一時ストレージ (Nova)	イメージ用ストレージ (Glance)	永続的ストレージ (Cinder)
OpenStackのシステムが利用するDB等の領域として利用 OpenStack DB 設定ファイル	仮想マシンが起動するOS用の領域として利用 	仮想マシンが起動するOS用の領域として利用 	仮想マシンに添付するボリュームとして利用 

OpenStack向けブロックストレージ

分散ストレージ

- 汎用IAサーバで安価に構成
- 2-3つの複製を作成
- スケールアウト型で容量と性能の拡張

EMC ScaleIO

Red Hat Ceph Storage

従来型ストレージ

- アプライアンスによる高い信頼性やバックアップ機能
- 重複排除等の高度なストレージの機能

NetApp FAS

EMC VNX

OpenStack環境でのストレージ選択ポイント

Red Hat Ceph Storage

- オープンソースのコストメリット
- S3互換のオブジェクトストレージも利用
- OpenStack環境での導入実績を重視

EMC ScaleIO

- 分散ストレージのコストメリットと共に通常ストレージと同様サポートが必要
- インストールや運用が容易

従来型ストレージ

- コストメリットより信頼性を重視
- 既存のストレージを活用したい

分散ストレージ　機能比較

比較項目	EMC ScaleIO(2.0)	Red Hat Ceph(1.3x&2.0)
データ複製数(default)	2面ミラー	3面ミラー
ストレージ多様性	ブロック	ブロック、オブジェクト、CephFS
ボリューム毎のQoS	○ IOPSもしくは帯域(KB/sec)で指定可能	×
RHEL-OSP環境に追加で必要な最少サーバ数	0台	3台 (ストレージノードは統合不可)
導入容易性	○　15分程度で容易にインストール	△ 従来のceph deployツールでは困難 →ver.2.0のAnsibleベースデプロイツール、RHOSP Directorによる改善に期待
GUI機能	○　Disk単位IO、リビルド・リバランス状況、通信帯域、ボリューム	△ ver.2.0のストレージコンソールGUIに期待
パフォーマンスチューニング	△　EMCから提供される方法のみ	○　管理者のスキル次第で深いところまで可能
メンテナンス時の性能劣化対策	○ メンテナンス時のリビルドの発生を抑制、メンテナンス後に書き込み差分のみ同期	×
実績	100ノード	4.7PB、シェアNo.1

OpenStackでストレージを利用する場合の疑問

OpenStackでストレージを利用する場合の疑問点

分散ストレージで十分なパフォーマンスは出るのか?

SnapShot取得時の性能劣化はあるのか?

HyperCoverged構成は取れるのか?

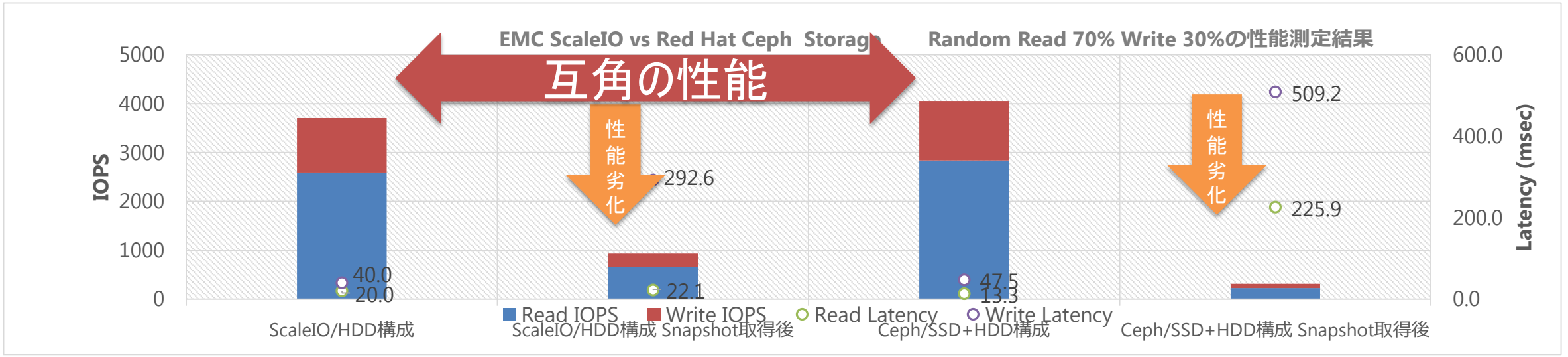
SSDの分散ストレージは有効か?

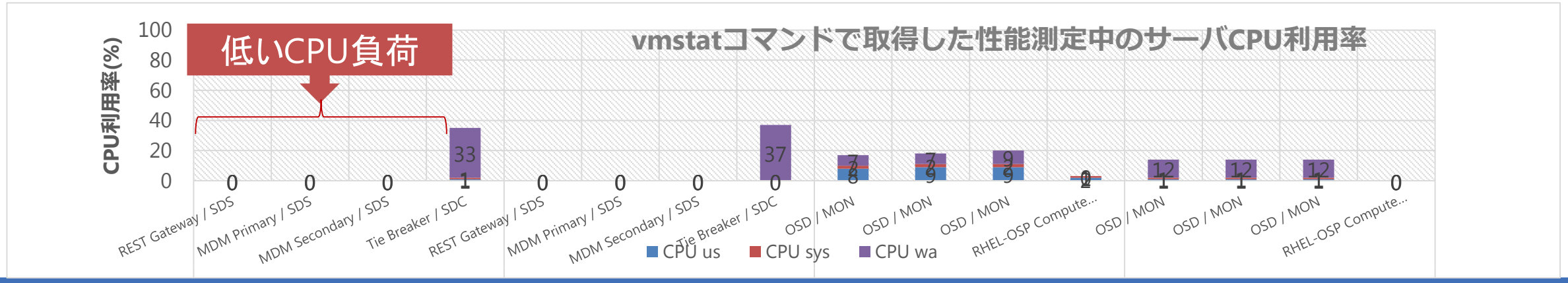
各分散ストレージの性能差はどのくらいか

実際に計測して疑問を解消

ScaleIO vs Ceph -Random Read/Write性能比較-HDD

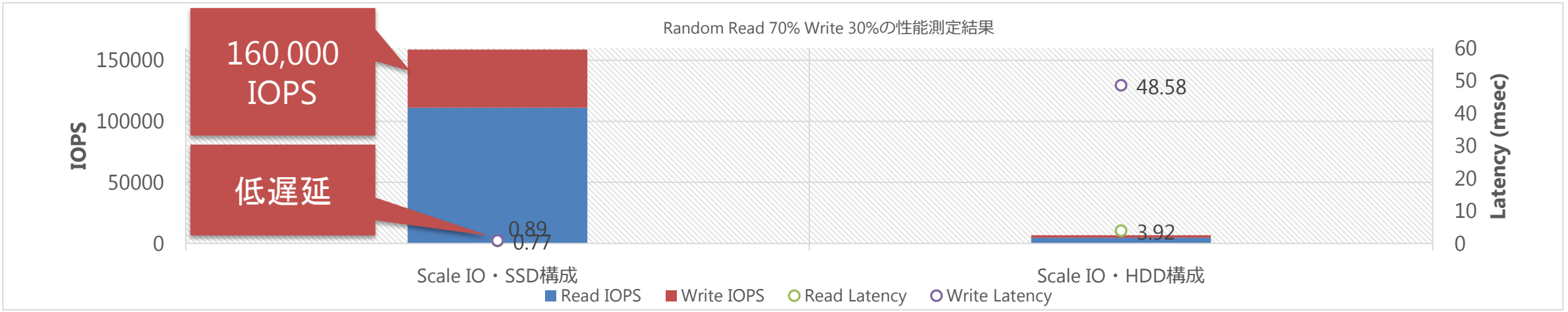
- CephとScaleIOでは、HDDで4000 IOPS程度でほぼ互角の性能
- SnapShotを取得すると性能劣化が起こる

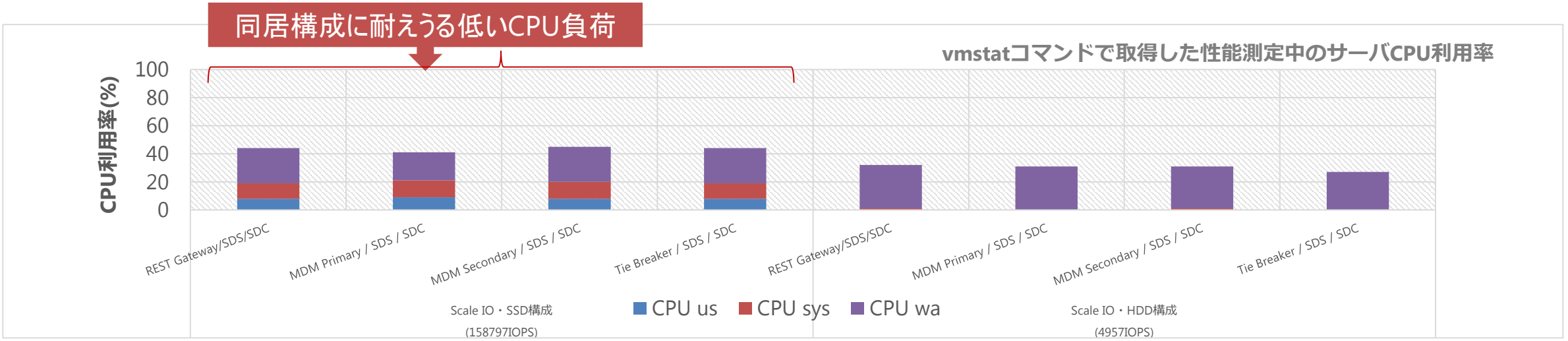




RHEL-OSP ScaleIO 同居構成 -Random Read/Write性能-

- SSDを利用すると、160,000 IOPSの性能を達成
- SSDを利用しての高負荷時にもVM+ストレージで実質20%程度のCPU負荷





計測からわかる様々な結果

実際の計測を行うと様々な結果がわかります。

分散ストレージ
でも十分な性能が出る

分散ストレージはSnap Shot後は
大幅な性能劣化

Hyper Converged構成は
ストレージのCPUの利用の
考慮が必要

SSDを使用すると高性能・
低遅延となり有効

HDD利用の同様な構成では
ほぼ互角の性能

実際に計測すると様々面が見えてくるので、疑問点の
計測は重要

ネットワンの考えるOpenStackストレージの最適解

データの性質	• データの増大スピードが速ければ「分散ストレージ」 • 高度なストレージ機能を重視するなら「従来型ストレージ」
実績、導入容易性	• 導入実績、多様性「Red Hat Ceph Storage」 • スモールスタート、サイジング不要、低オーバーヘッド、拡張時や障害時やメンテナンス時の性能劣化を抑えたいなら「EMC ScaleIO」
ストレージの性能	• スナップショットを取得するボリュームはスナップショットオフロードストレージの使用を検討 • 分散ストレージでスナップショットを使用するなら、削除する運用を検討

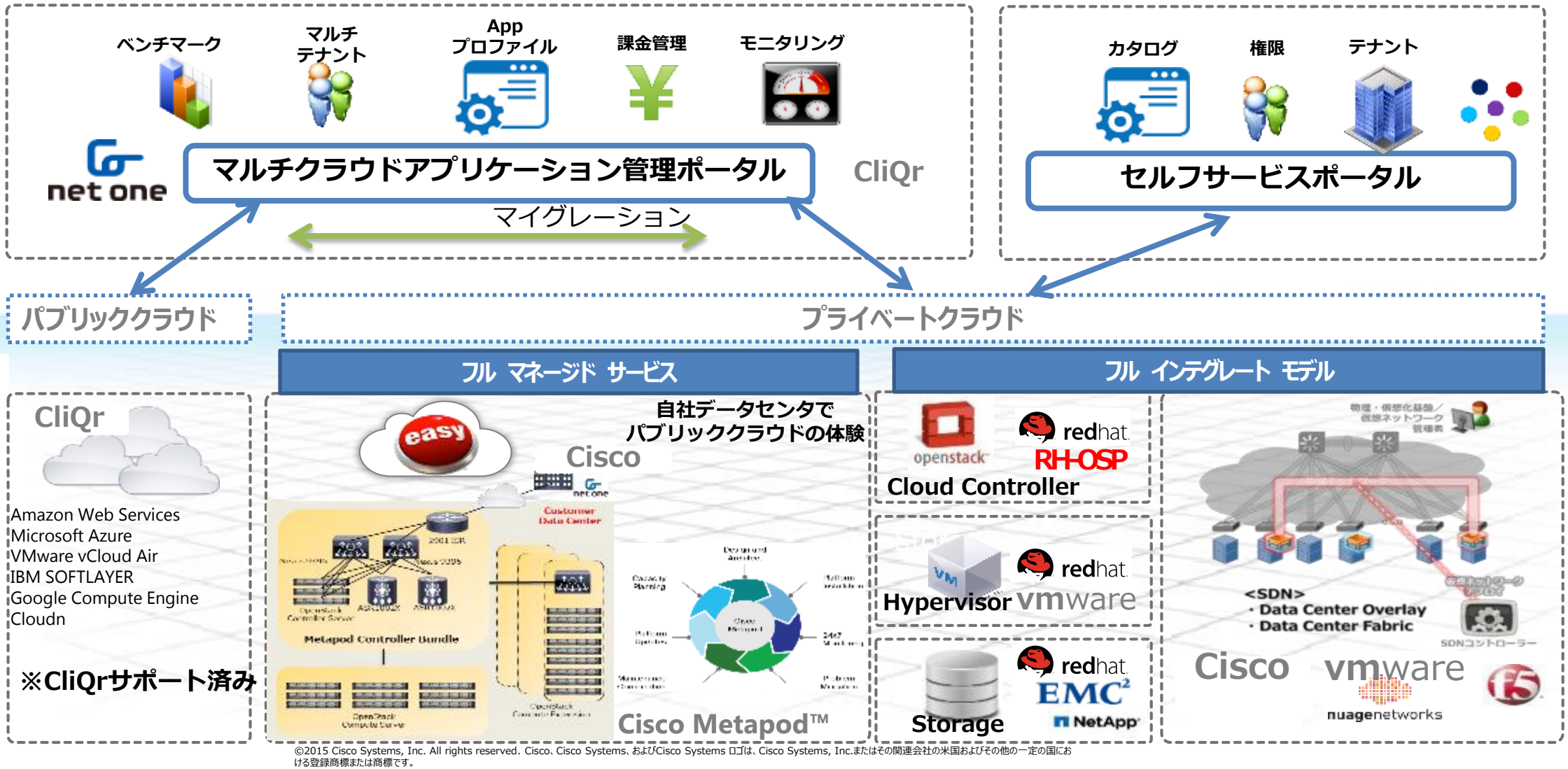
検証の結果とWhite Paperについて

検証の詳しい結果はEMC、レッドハットと共同執筆したWhite Paperに記述致しました

ネットワンブースにお越しください。

本セッションの内容を盛り込んだホワイトペーパーを差し上げます。

ネットワンの提供するクラウド



つなぐ ∠ むすぶ ∠ かわる

net one