

使ってみなければわからない！～クラウド時代の”XX”シリーズ～  
第1回：クラウド時代のストレージとは？

マルチベンダであるネットワンが考える、  
ストレージ製品の用途別比較

2018年10月31日  
ネットワンシステムズ 株式会社  
東日本第1事業本部 第2営業部



## ■■■ 本セッションのアジェンダ ■■■

1. クラウド時代のストレージ要件
2. クラウドを鑑みた使い分け
3. プライマリストレージを考える
4. セカンダリストレージを考える
5. ストレージ観点でDRを考える
6. 新トレンド
7. おわりに

# はじめに

**本日はご来場いただき頂きありがとうございます。**

本セッションではデータ種別による使い分けから各カテゴリに対するストレージを考えます。

## ポイント

1

**クラウドやIoTといった新要素の登場により、変化したことについてお話させていただきます。**

- クラウドの登場により、変化した要件について
- プライマリストレージ、セカンダリストレージというような区分け
- 平時のDRの考え方の変化

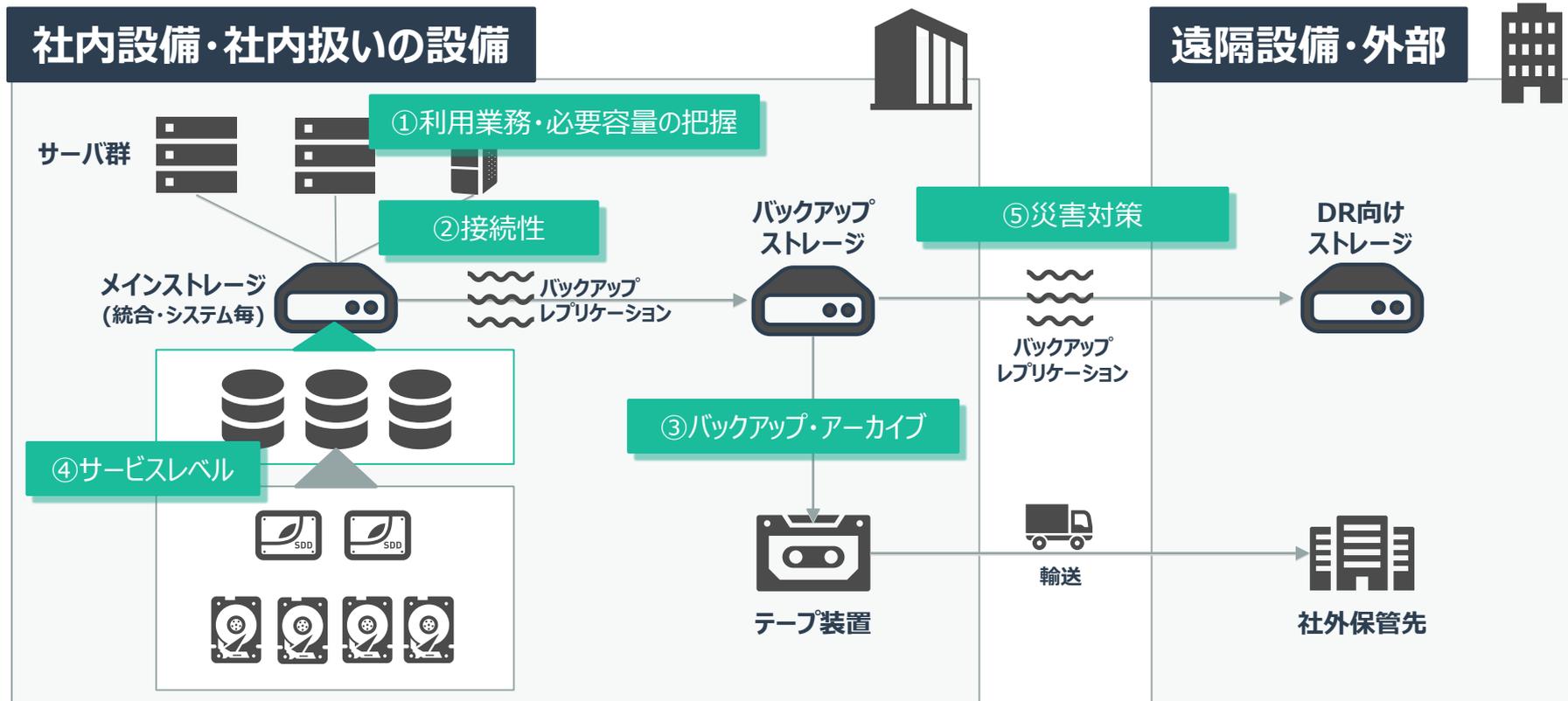
2

**現状を否定するものではありません。**

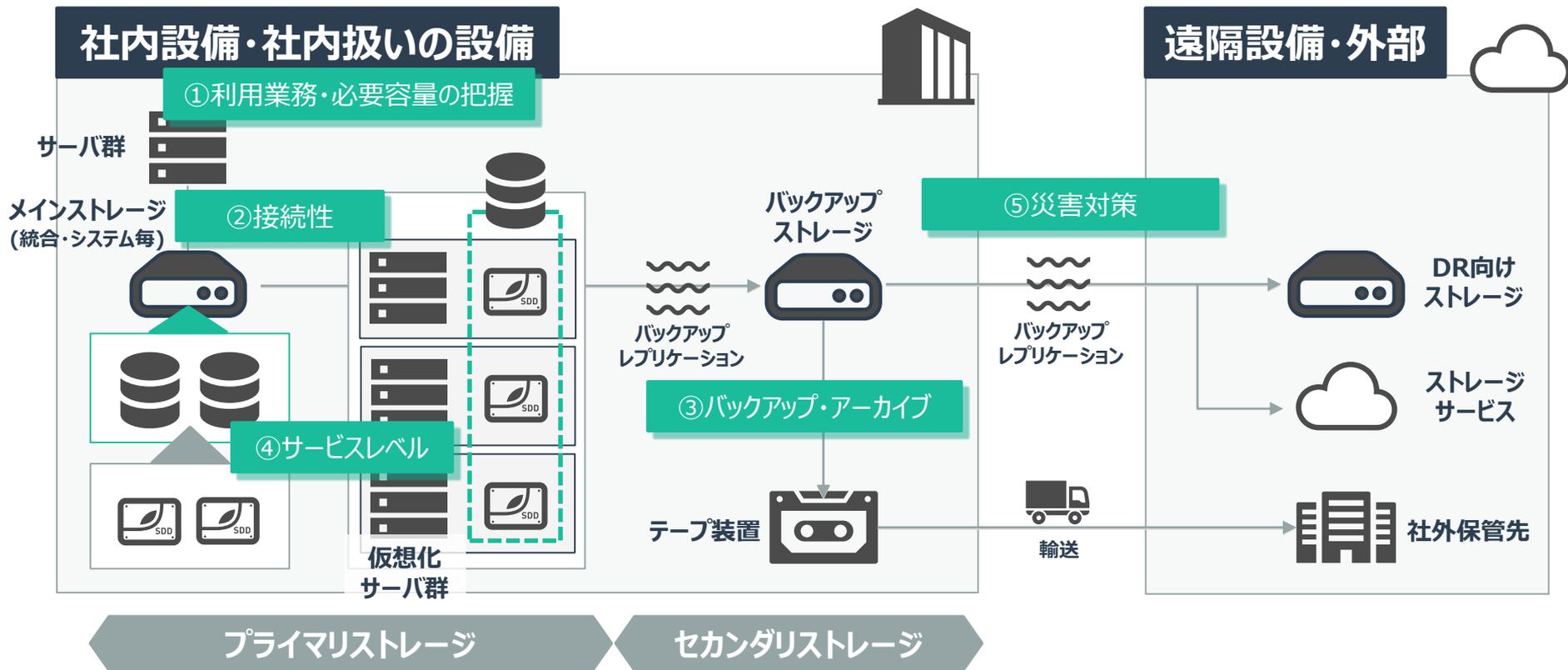
- 新しい要素の登場で変わった観点について述べております。

# クラウド時代のストレージ要件

# 🔗 検討時の要件（今まで）



# 📍 検討時の要件（これから）



# 要件変化の対比

	要件（従来）	要件（これから）
利用業務・必要容量の把握	用途を洗い出して予め設置 必要な容量を必要だけ予めきっちり確保	使いたいときに使いたいだけ 必要な時に必要な容量を必要だけ使いたい <b>コスト低減</b>
接続性	SAN・NASの利用	SAN・NAS利用 + 考慮したくない <b>運用工数低減</b>
バックアップ・アーカイブ	バックアップ専用装置・NAS利用	より大容量でかつ安価 <b>コスト低減</b>
サービスレベル	筐体の牢固性とディスク階層	より高速でかつ落ちない <b>SLA向上</b>
災害対策	遠距離センターへのミラー	被災時のみの稼働とコスト <b>コスト低減</b>

クラウドの発展やそれに纏わる新しいテクノロジーの登場、既存領域においてもテクノロジーの進歩により、従来の制約に縛られない新しい考え方で更なるコスト・運用工数の低減、SLAの向上が可能に

# クラウドを鑑みた使い分け

# クラウドを鑑みた使い分け

## プライマリストレージ

### SORシステム向け



Block



1

## セカンダリストレージ

### SOEシステム向け

現状は非常にサイロになっているケースが多くDXが生まれにくい



文書ファイル群

File



3



Webなどのコンテンツ群

Object



2



システムログ センサーログ

BIG-DATA



システム・APP  
バックアップ アーカイブログ  
REDOログ

Backup



DR

ビジネスの差別化を行うに当たり  
この領域にAIを積極的に取り入れることが必要

①プライマリ、②セカンダリ、③DR毎に、**使い分け**や**要件への適合性**を確認していきます。  
残念ながら本日は時間の関係もあり、**新要素が含まれるもの**を中心にご紹介をさせていただきます。

# ストレージの種別

	特徴・機能	課題・デメリット	
レガシー	<ul style="list-style-type: none"><li>● DAS, SAN, NAS のシステム専用のストレージ</li><li>● システムで個別設計しパフォーマンスの最適化を実現する</li><li>● 成熟した管理ツール</li><li>● 成熟した機能 (Auto tiering, Snap, replication, de-duplication, compression)</li></ul>	<ul style="list-style-type: none"><li>● 機器コストが高い</li><li>● 運用保守コストが高い</li><li>● スケールアップでの拡張の為、拡張限界があり高コスト</li></ul>	機密性の高い 高価な小規模データ RPO/RTO重視
アプライアンス	<ul style="list-style-type: none"><li>● ハードウェアとソフトウェアを組み合わせたアプライアンスで提供</li><li>● 専用HWにチューニングを行いパフォーマンスを最適化</li><li>● 簡単なインストール・管理ツール</li><li>● スケールアウト型の拡張に対応</li></ul>	<ul style="list-style-type: none"><li>● 機器コストは安くはない</li><li>● 運用・保守コストは安くはない</li><li>● スケールアウト型の拡張に対応している場合が多いが拡張上限が限られる場合がある</li></ul>	機密性の高い 小規模データ 拡張性重視 管理性重視
Software-Defined	<ul style="list-style-type: none"><li>● ソフトウェアベースのストレージ</li><li>● ソフトウェアにコモディティハードウェアを組み合わせて構成</li><li>● 次世代のアーキテクチャ(オブジェクトストレージ)</li><li>● データにアクセスする度に費用は発生しない</li><li>● コストを抑えてスケールアップが可能</li></ul>	<ul style="list-style-type: none"><li>● サービスシステムを組み上げる為にはサービス提供用のソフトウェアが動くサーバ群と組み合わせる必要がある</li><li>● サービスを組み上げる為に開発が必要になる場合がある</li></ul>	機密性の高い 大容量データ 拡張性重視 コスト重視
Public Cloud	<ul style="list-style-type: none"><li>● 機器資産を持たず利用量に応じたコストを清算</li><li>● 低価格でのサービスの利用が可能</li><li>● 必要に応じてサービス契約を増やすことで スケールアップ可能</li></ul>	<ul style="list-style-type: none"><li>● Public Cloud に機密データを配置することになる</li><li>● データにアクセスする度に費用が発生する場合がある</li></ul>	機密性の低い 安価なデータ 外部アクセス重視

# 対応製品群

## 対応製品群とプライマリ・セカンダリにおける論点

		レガシー	アプライアンス	Software-Defined	Public Cloud	
プライマリ	ブロックベース	<ul style="list-style-type: none"> <li>Pure Storage Flash Array</li> <li>NetApp FAS</li> <li>EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>EMC VxRail</li> </ul>	<ul style="list-style-type: none"> <li>VMware vSAN</li> </ul>	<ul style="list-style-type: none"> <li>AWS EBS</li> <li>Azure Storage(Block)</li> </ul>	各種類の使い分けについて <b>1</b>
	ファイルベース	<ul style="list-style-type: none"> <li>NetApp FAS</li> <li>EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>EMC Isilon</li> </ul>	<ul style="list-style-type: none"> <li>ONTAP Select</li> <li>Unity VSA</li> <li>IsilonED Edge</li> </ul>	<ul style="list-style-type: none"> <li>ASW EFS</li> <li>Azure Storage(File)</li> </ul>	
セカンダリ	オブジェクトベース	—	<ul style="list-style-type: none"> <li>Pure Storage Flash Blade</li> <li>EMC ECS</li> </ul>	<ul style="list-style-type: none"> <li>Scality RING</li> </ul>	<ul style="list-style-type: none"> <li>AWS S3</li> <li>Azure Storage(BLOB)</li> </ul>	セカンダリの選び方について (FS vs OBJ) <b>2</b>

# ①プライマリストレージを考える

# プライマリストレージ対象

		レガシー	アプライアンス	Software-Defined	Public Cloud	
プライマリ	ブロックベース	<ul style="list-style-type: none"> <li>● Pure Storage Flash Array</li> <li>● NetApp FAS</li> <li>● EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>● EMC VxRail (正体はvSAN)</li> </ul>	<ul style="list-style-type: none"> <li>● VMware vSAN</li> </ul>	<ul style="list-style-type: none"> <li>● AWS EBS</li> <li>● Azure Storage(Block)</li> </ul>	各種類の使い分けについて <b>1</b>
	ファイルベース	<ul style="list-style-type: none"> <li>● NetApp FAS</li> <li>● EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>● EMC Isilon</li> </ul>	<ul style="list-style-type: none"> <li>● ONTAP Select</li> <li>● Unity VSA</li> <li>● IsilonED Edge</li> </ul>	<ul style="list-style-type: none"> <li>● ASW EFS</li> <li>● Azure Storage(File)</li> </ul>	
セカンダリ	オブジェクトベース	—	<ul style="list-style-type: none"> <li>● Pure Storage Flash Blade</li> <li>● EMC ECS</li> </ul>	<ul style="list-style-type: none"> <li>● Scality RING</li> </ul>	<ul style="list-style-type: none"> <li>● AWS S3</li> <li>● Azure Storage(BLOB)</li> </ul>	



確かめること

- ALL Flashは本当に速いのか
- HCI/vSAN・クラウドストレージは使い物になるのか

# ALL Flashアレイは何かいいのか

1

## 記憶デバイスが圧倒的に速い

All Flashに使われているeSSDにより、圧倒的な性能を誇りながらも予備領域を大きくとっておくことで書き込み寿命を大幅に改善

2

## インライン重複排除・圧縮による格納データ量の削減

従来のプライマリストレージでは提供されていなかった「インライン重複排除」、「圧縮機能」の提供が一般的予備領域を大きくとっておくことで書き込み寿命を大幅に改善

3

## 省スペース・省電力

従来のHDDベースのように、ストレージの性能の向上のためにHDDの数を増やす必要がないため、ラック単位のスペースが必要だったストレージを数Uのスペースに収めることができ、データセンターへの利用料や電力コストを大幅に削減

4

## システム設計や運用をシンプル化

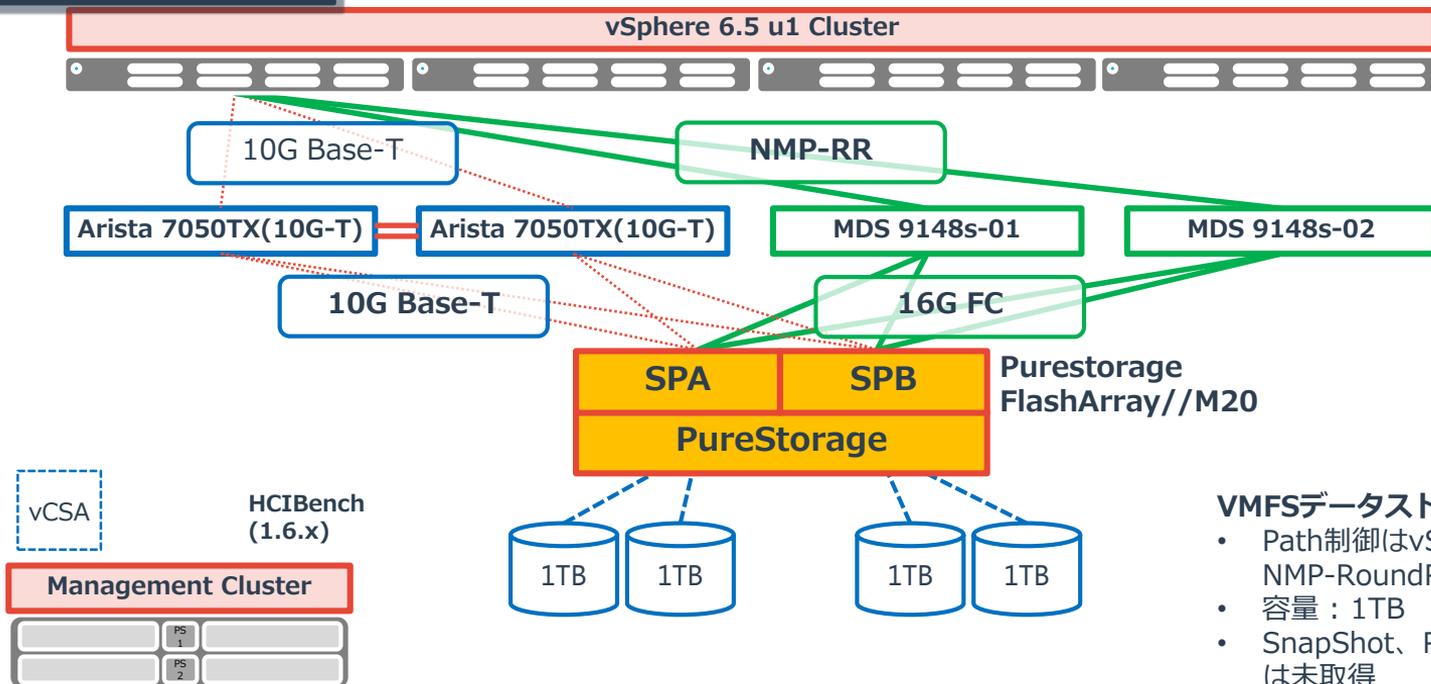
I/O性能を引き出すためのアプリケーションでのチューニングや、アライメント調整といったストレージとの最適化が不要

デバイス	IOPS (1本あたりの論理値)
SATA(7200rpm)	80 IOPS
SAS(1000rpm)	140 IOPS
eSSD	10,000 IOPS

# Block (VMFS DataStore) 検証環境構成

## 検証環境構成

(Purestorage FlashArray//M20)



### VMFSデータストア条件

- Path制御はvSphere NMP-RoundRobin
- 容量：1TB
- SnapShot、Replicationは未取得

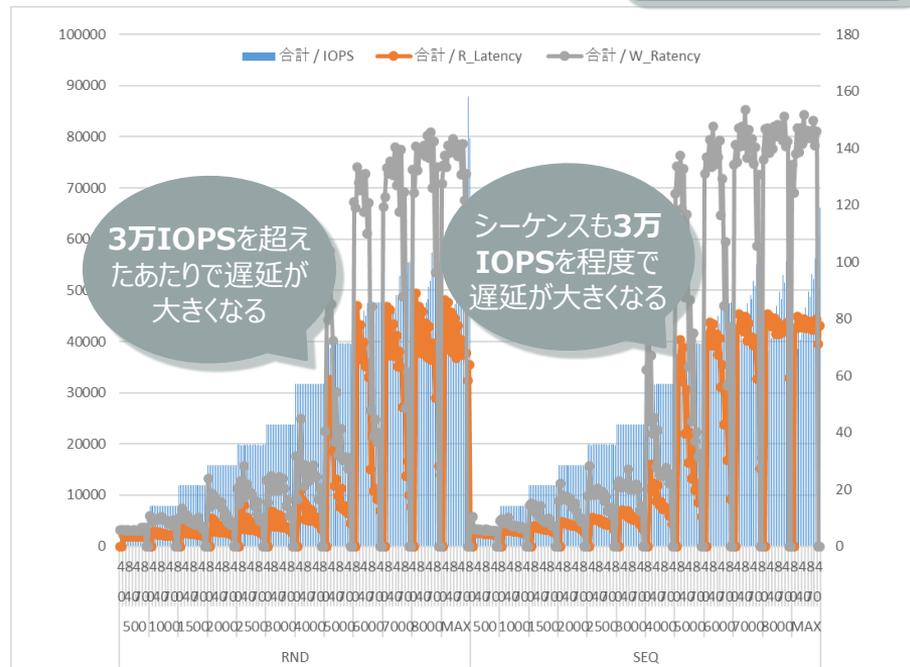
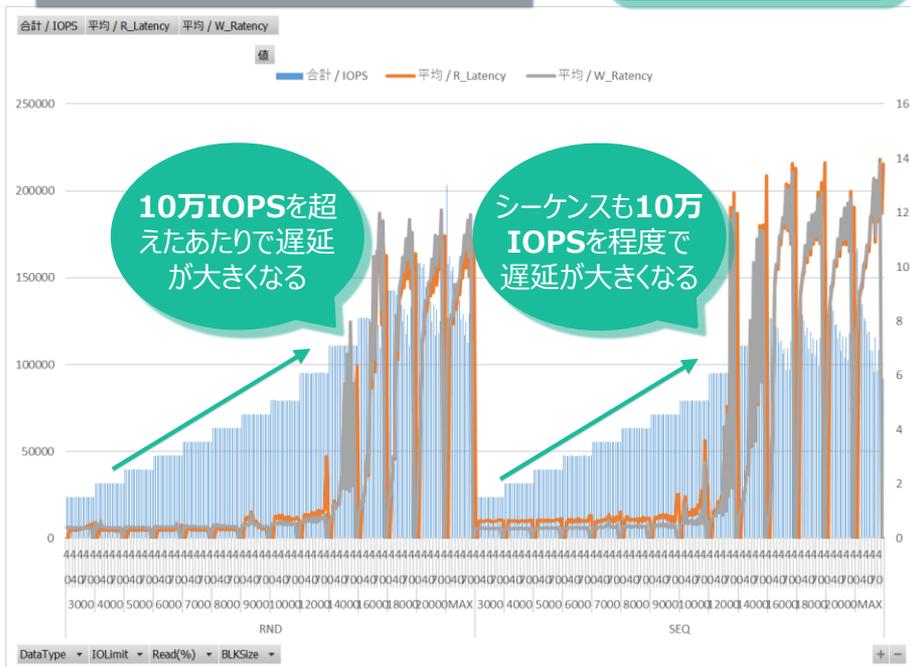
# ALL Flash アレイの性能は本物か

## 検証結果

(Purestorage FlashArray//M20)

FlashArray//M20

ハイブリッドストレージ



同環境でのパフォーマンス試験において**3倍超**のIOPS性能を発揮

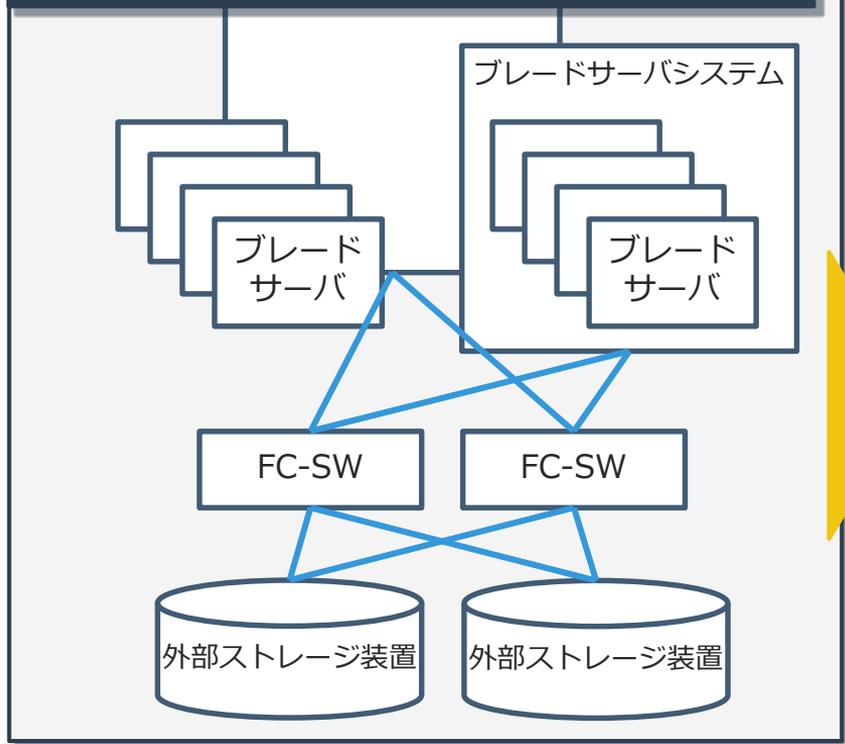
# ALL Flash アレイの性能は本物か

【ご参考】 限界性能 (Purestorage各モデル)

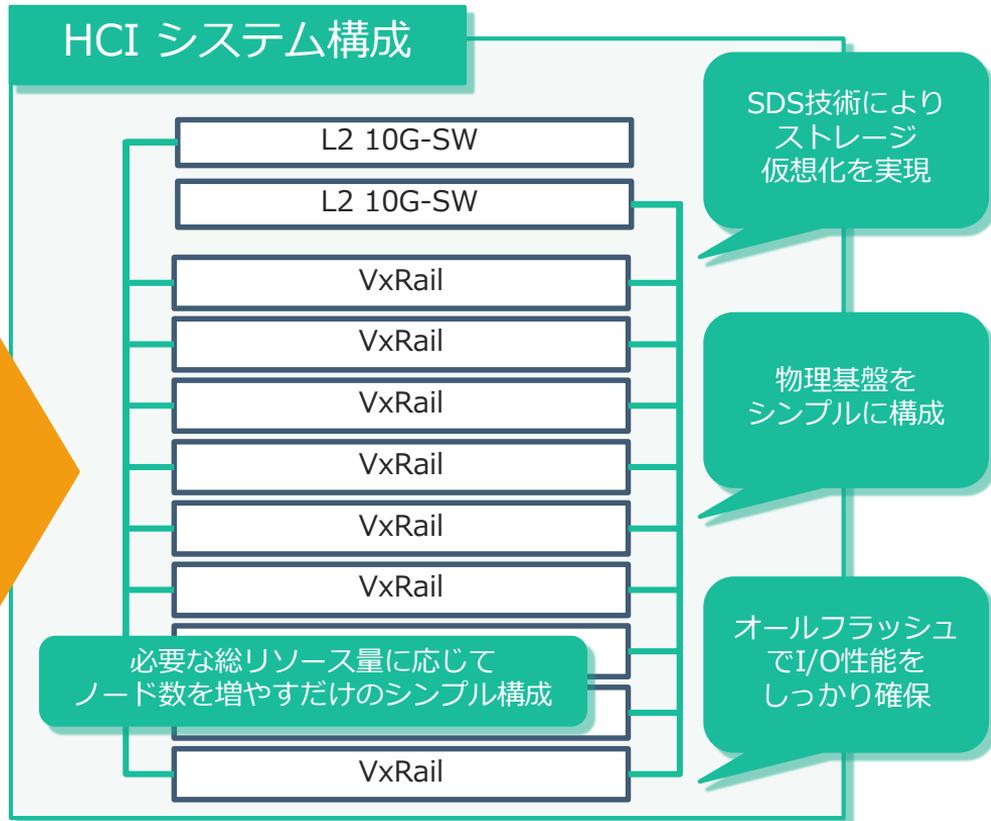


# HCIのおさらい(1)

## 従来のサーバ+ストレージのシステム構成



## HCI システム構成



# HCIのおさらい(2)

## 従来型システムに対する管理コンセプト優位性

	従来型 (サーバ+ストレージ)	HCI
特徴	個別コンポーネント (サーバ、FC-SW、ストレージ)	クラスタ・ノード単位
インベントリ管理	製品毎 (サーバ、FC-SW、ストレージ)	一部統合管理 (サーバインフラへの集約)
障害時	手動復旧 (冗長化の縮退、システムの縮退)	自動復旧 (別ノードへコピーし冗長化を維持)
拡張時	設計要素多く リードタイム長い	設計要素少なく リードタイム短い

ノード単位で増設  
ハードウェアレイヤから  
一括した管理が可能

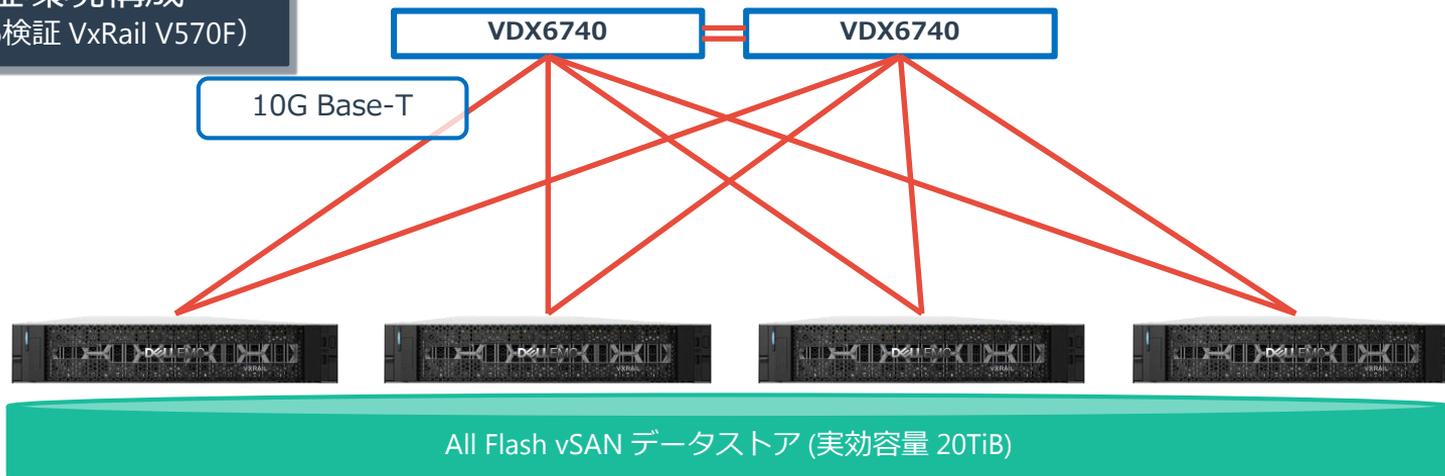
外部ストレージや  
FC-SAN機器の  
管理から解放

SDS技術により  
データの再配置を  
行い冗長性を  
保ちます

すぐ誰にでも  
シンプルな手順で  
増設可能

# HCIのストレージ性能は本物か

検証環境構成  
(vSAN6.6検証 VxRail V570F)

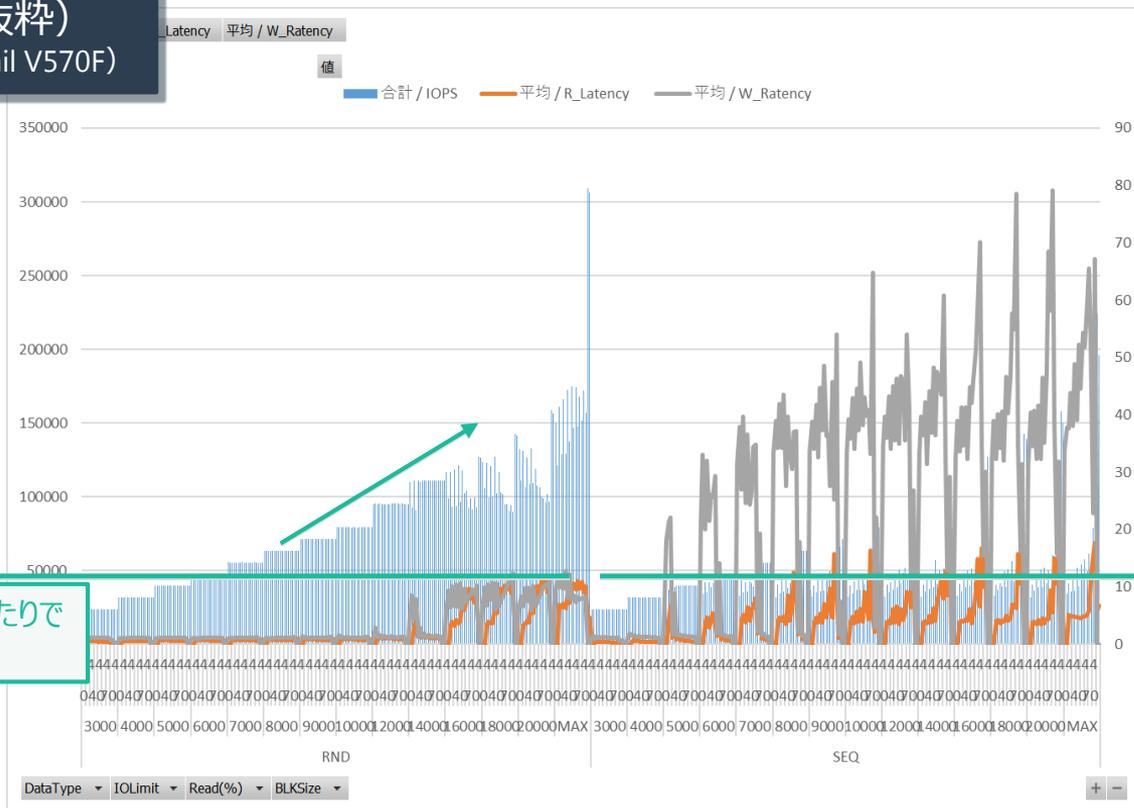


## 【vSphere 6.5 u1 + vSAN 6.6.1】

- VxRail V570F (Dell PowerEdge R740XDベース)
  - CPU Xeon 6130 2.1G 16C \* 2S
  - Mem 768GB (32GB \* 24)
  - 2port 10G-SR
  - Boot領域 M.2 PCIe接続 SSD
  - キャッシュSSD : 400GB \* 1本、キャパシティSSD : 1.92TB \* 3本

# HCIのストレージ性能は本物か

検証結果 (抜粋)  
(vSAN6.6検証 VxRail V570F)

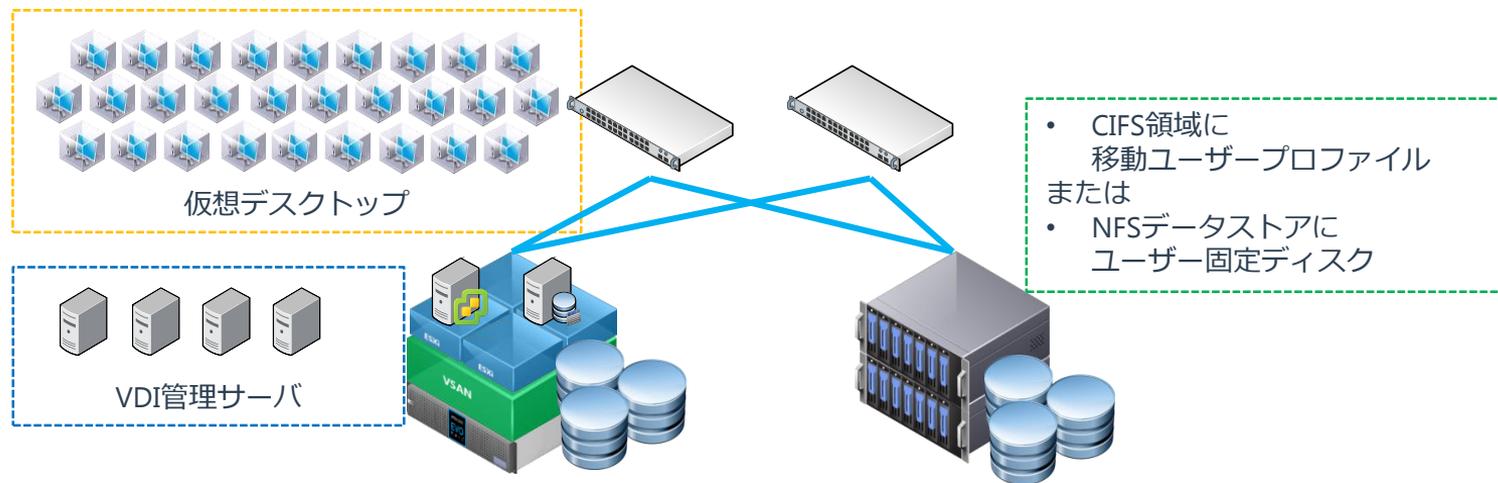


10万IOPSを超えたあたりで  
遅延が大きくなる

シーケンシャルアクセスについて  
は、ランダムに比べ上昇せず

# VDI環境のサーバーリソースとしての利用

VDIなどの業務開始・終了時に突発的なIOが集中する傾向のあるシステム領域の負荷をVSANデータストアで対応し、大容量、かつアーカイブ用途の傾向が強いユーザーのプロファイル領域を安価な外部ファイルサーバで組み合わせる事で最適化が可能です。



## VSANデータストア（システム領域用）

キャッシュヒット率の高いシステム領域のIO特性を活かして、ブートストーム、ログインストームなどのVDI特有のIO負荷を、SSDの高速なR/W処理でカバーする事が可能。  
VDI基盤のサーバー構成をAll in Oneで構成して運用をシンプル化。

## 外部ファイルサーバー（大容量NAS）※推奨10G Ether

ユーザー用途に応じて外部に大容量のNASを安価に用意する事を推奨。  
要件により容量・性能は個別サイジング

- NFSデータストアをESXiにマウントするパターン
- CIFSファイルサーバをVDIから移動プロファイルで接続

# クラウドブロックストレージ(例 : EBS)

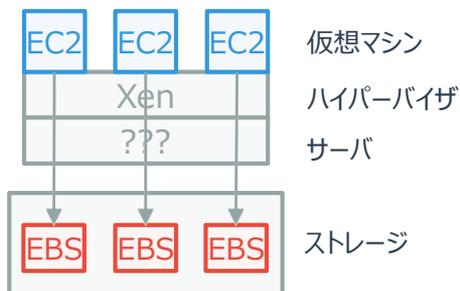
## 概要

EC2にマウントする、ブロックストレージ。

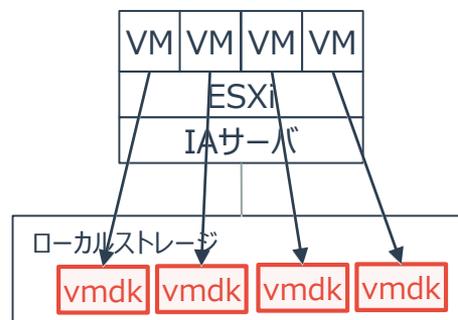
## ユース ケース

EC2に非揮発性のディスクをマウントしたい  
**IOPS保証されたディスクがほしい**

## AWS デザイン



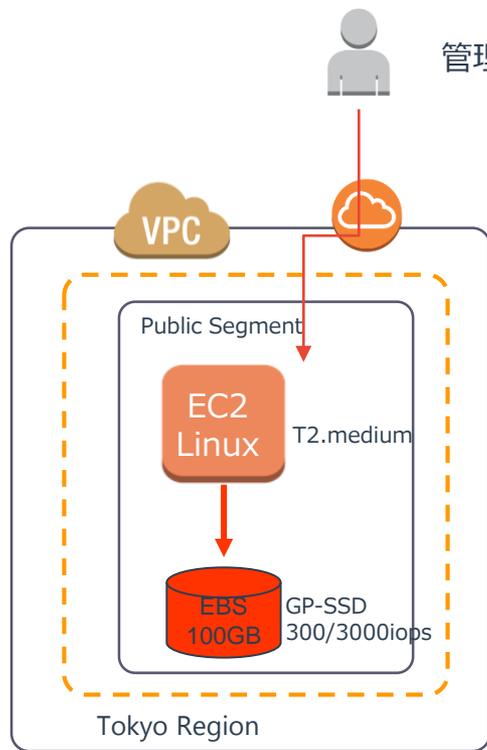
## 一般的なデザイン



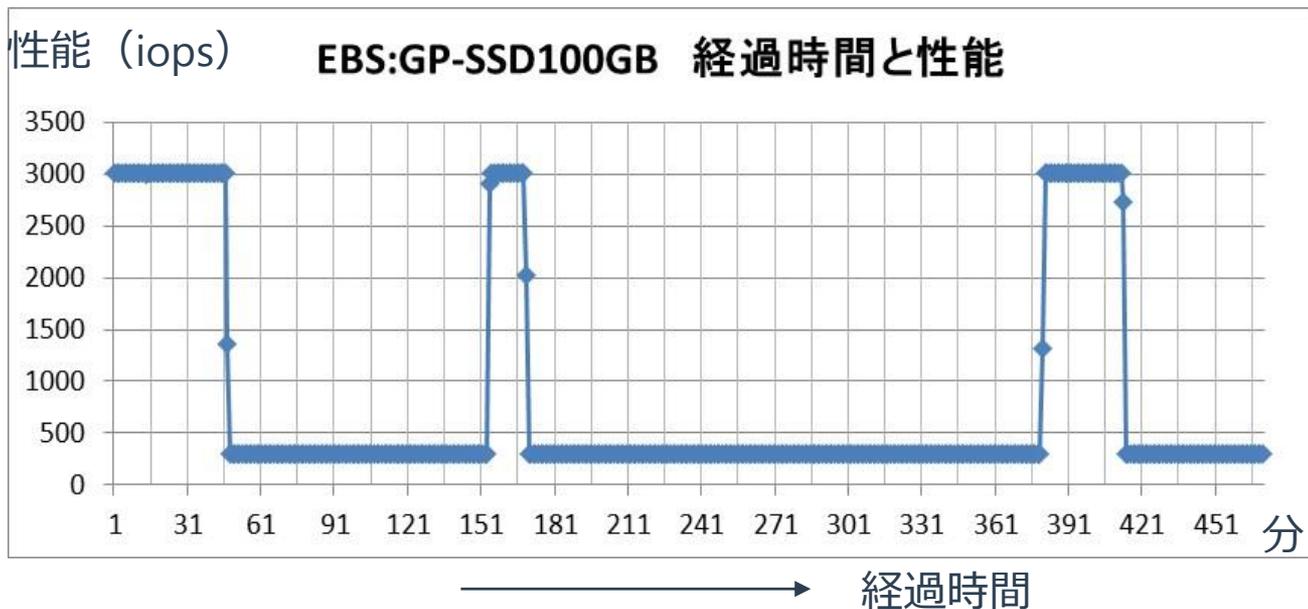
# スペック通りのスペックを示すか？－検証項目

カテゴリ	検証項目	内容	check	Done
EBS	EBS100Gの基本性能	GP-EBSの100G、バースト3000iops、基本300iopsのボリュームに対して、EC2インスタンスタイプ、基本的な性能をもつt2.mediumのシングルインスタンスで、4KB~1MBまでの性能を取得する。バーストが許容されている時間帯でテストする。IOPS、MB/s、response time	✓	
	EBS100Gの基本性能	GP-EBSの100G、バースト3000iops、基本300iopsのボリュームに対して、EC2インスタンスタイプ、基本的な性能をもつt2.mediumのシングルインスタンスで、4KB~1MBまでの性能を取得する。バーストが許容されなくなった時間帯でテストする。IOPS、MB/s、response time（項番.2と比較する	✓	
	EBS2TBの基本性能	t2.mediumのシングルインスタンスで、4KB~1MBまでの性能を取得する。	✓	
	EBS100Gのバースト特性	GP-EBSの100G、バースト3000iops、基本300iopsのボリュームに対して8kbRanReadのio負荷を継続して、経過時間とiopsの関係を確認する。想定結果として、30分程度で基本性能に低下する。	✓	
	EBS2TBのバースト特性	GP-EBSの2TB、基本6000iopsのボリュームに対して8kbRanReadのio負荷を継続して、経過時間とiopsの関係を確認する。想定結果として、時間経過に依存しない。	✓	
	P-EBS 340G	10000iopsのボリュームに対して、EC2インスタンスタイプ、基本的な性能をもつm4.xlargeのシングルインスタンスで、4KB~1MBまでの性能を取得する。IOPS、MB/s、response time	✓	
	P-EBS 340G	10Gbpsに対応したc4.8xlargeのシングルインスタンスで性能を取得する。IOPS、MB/s、response time	✓	

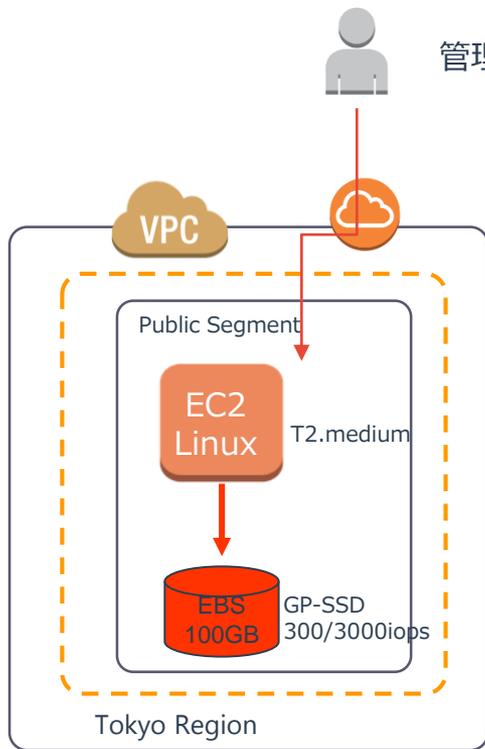
# EBSバースト特性の確認結果



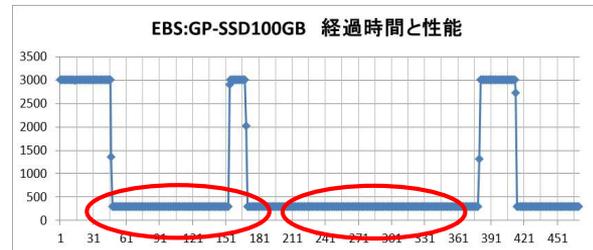
300IOPSの基本性能に対して、4時間程度毎に、30分程度バースト性能として3000IOPS利用可能なことを確認



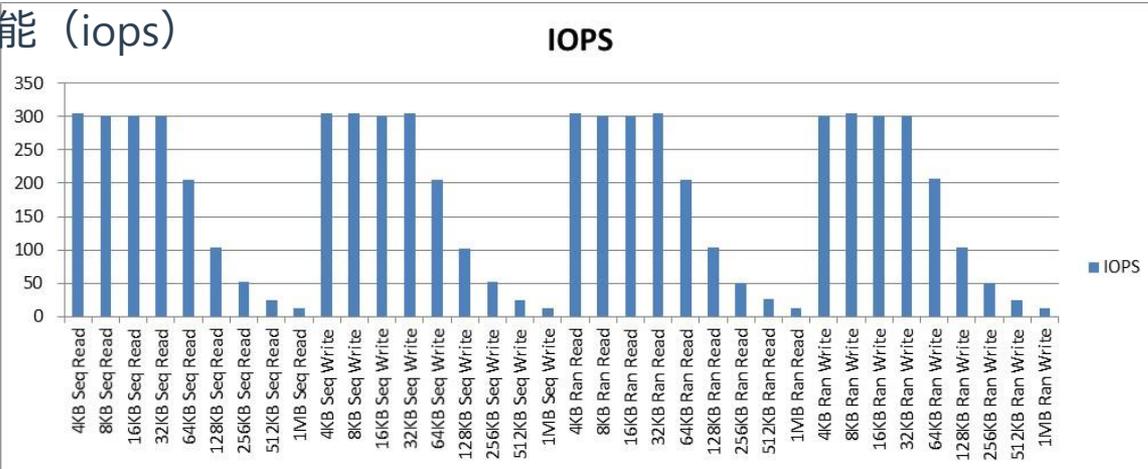
# EBS 性能がキャッピングされているときの性能



## 300IOPSのブロック特性

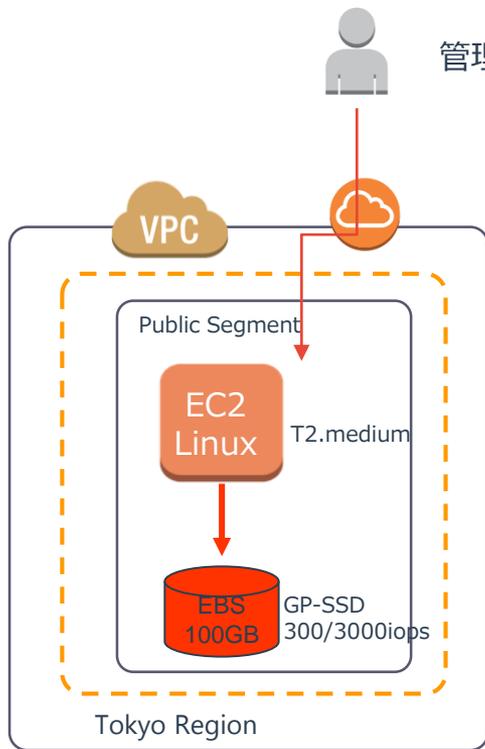


## 性能 (iops)

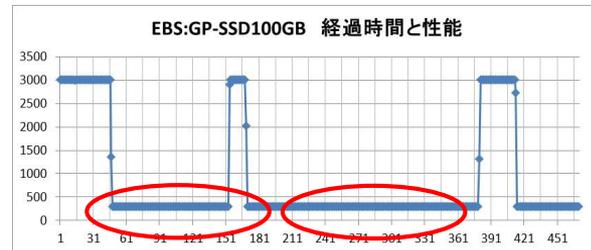


## ブロックサイズ

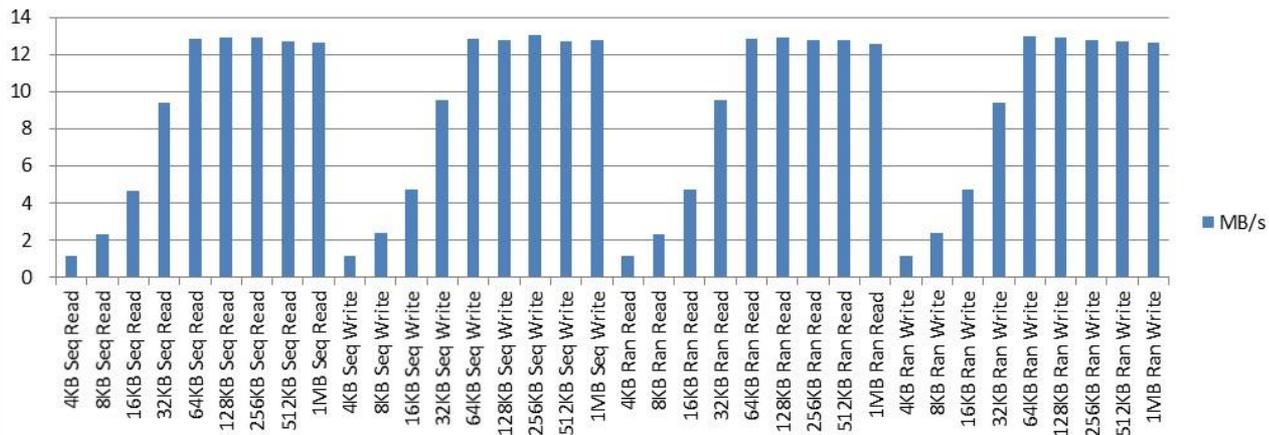
# EBS 性能がキャッピングされているときの性能



300IOPSの利用可能帯域

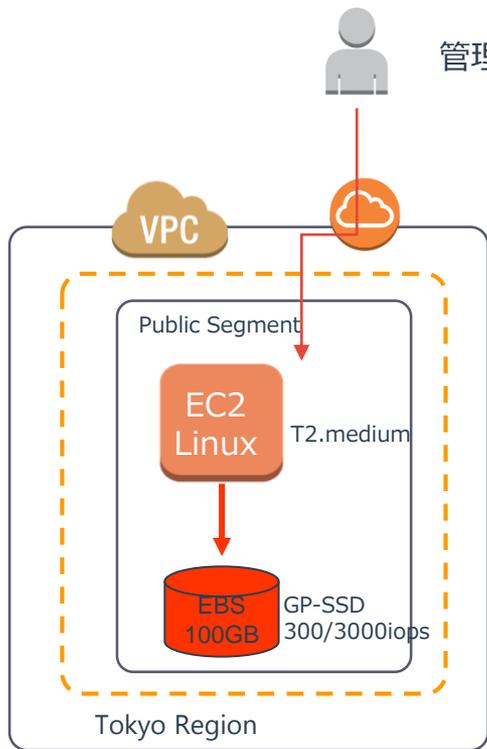


帯域 (MB/s)

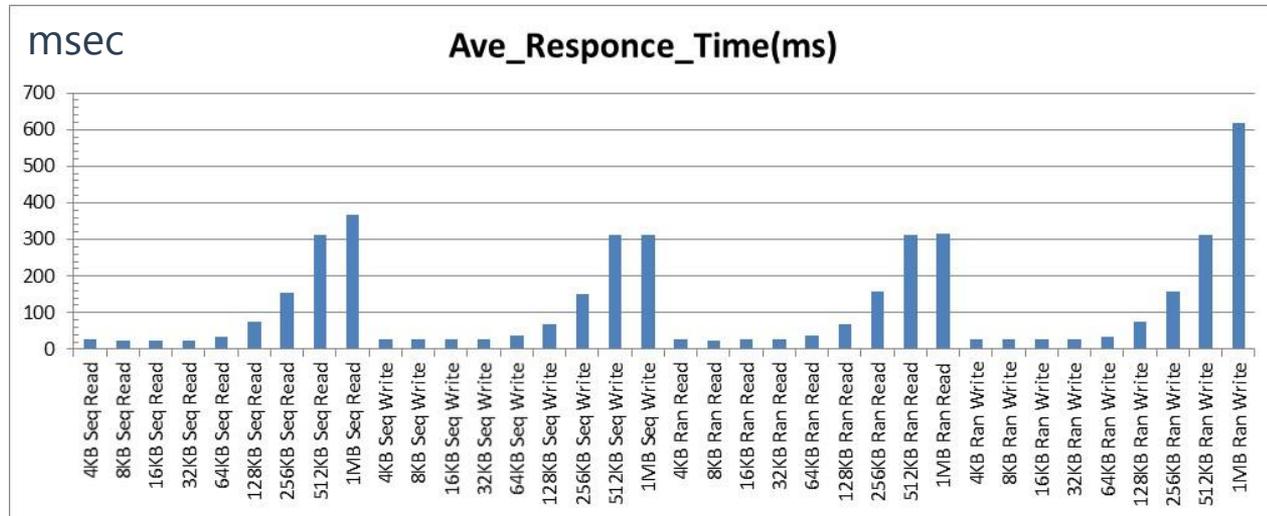
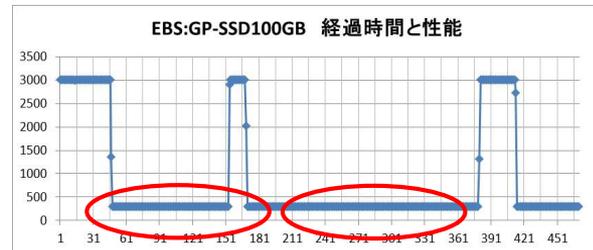


ブロックサイズ

# EBS 性能がキャッピングされているときの性能



300IOPSのレスポンスタイム



ブロックサイズ

# EBSパフォーマンス検証のまとめ

- GP-SSD（ジェネラルパーパスSSD）は、1GBあたり3IOPSが基準になっています。100GBのボリュームを作成すると、300IOPSが基準となります。
- IOPSによるキャッピングに加えて、帯域による上限が定義されている振る舞いがあります。
- GP-SSD 100GBの場合は、13MB/sの上限が定義されているような振る舞いとなりました。
- 300IOPSにキャッピングされている際の、レスポンスタイムは、8KBでおよそ25msecでした。

# 比較検討-構成面-

観点	レガシー	Software-Defined	クラウド
性能	ALL Flashは現在考えられる最大の性能	フラッシュ活用で最大近くまで高めることができる	選定サービスによる
拡張性	スケールアップ	スケールアウト	構成意識せず柔軟
可用性・信頼性	専用設計、専用HWにより、最高の可用性・信頼性	ソフトウェア主導でHWに依存せずシステム全体で担保	選定サービスによる
自由度 (カスタマイズ)	物理環境依存	SW定義によるポリシー制御	選定サービスによる
初期設計	既存の考え方踏襲可	左記に加えストレージポリシー設計が必要	既存システムの連携を
初期構築/移行	既存環境を踏襲した構築。コンピュートの移行で完結	左記に加えストレージ移行追加	クラウド上のリソースは迅速に

# 比較検討-運用面-

あくまで相対での評価になります

観点	レガシー	Software-Defined	クラウド
ユーザ向け提供サービス	リソース提供 リソース払い出し時は各コンポーネントキャパシティ考慮の上払い出し 	柔軟に払い出し可能/ポリシー制御による自動化も可 	柔軟にリソース提供可能 
管理者向け運用管理	サービス維持管理 HWに依存のため、柔軟なサービスレベル定義が困難 	ストレージ階層もSW定義によって柔軟にサービスレベルを定義可能 	SW定義によるサービスレベル選択可能 
	構成管理 専用ソフトなどが必要な場合あり 	外部ストレージ特有の構成管理は省略可 	外部ストレージ特有の構成管理は省略可 
	稼働管理 複数コンポーネントにまたがる管理が必要 	仮想/物理統合的に管理するには別途ツールが必要 	オンプレ環境同様 
	障害管理 (HW) Storageに向けた対応体制などが必要 	HWストレージとしては削減可能 	サービス提供元へアウトソース可能 
	保守管理 専用機のため特化した保守サービス 	HWアプライアンスの為、比較的保守は簡易 	サービス提供元へアウトソース可能 

# 「これから」の要件への適合性

	要件（これから）	レガシー	SD	クラウド
利用業務・ 必要容量の把握	使いたいときに使いたいだけ 必要な時に必要な容量を必要なだけ使いたい	△	×	○
接続性	SAN・NAS利用 + 考慮したくない	△	○	○
バックアップ アーカイブ	より大容量でかつ安価	—	—	—
サービスレベル	より高速でかつ落ちない	○	○	○
災害対策	被災時のみの稼働とコスト	△	△	○

# まとめ

各カテゴリ間は一長一短であり優先度の高い要件で選択

運用	詳細な障害調査(根本原因の早期発見)	レガシー
機能	(業務要件から)純粋に性能が必要	
機能	三重化以上の冗長性が必要	
コスト	コンピュータリソースが独立しており、二重投資	
運用	ストレージの運用を簡略化したい	Software-Defined
機能	ソフトウェアにて実現している機能で柔軟に対応したい	
機能	自動化をスムーズに行いたい (SW実装+コンポーネント減)	
コスト	使いたいときに使いたいだけ使いたい	クラウド
運用	基盤運用をオフロードしたい(IaaSの場合)	

## ②セカンダリストレージを考える

# セカンダリストレージ対象

		レガシー	アプライアンス	Software-Defined	Public Cloud	
プライマリ	ブロックベース	<ul style="list-style-type: none"> <li>Pure Storage Flash Array</li> <li>NetApp FAS</li> <li>EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>EMC VxRail</li> </ul>	<ul style="list-style-type: none"> <li>VMware vSAN</li> </ul>	<ul style="list-style-type: none"> <li>AWS EBS</li> <li>Azure Storage(Block)</li> </ul>	各種類の使い分けについて <b>1</b>
セカンダリ	ファイルベース	<ul style="list-style-type: none"> <li>NetApp FAS</li> <li>EMC Unity</li> </ul>	<ul style="list-style-type: none"> <li>EMC Isilon</li> </ul>	<ul style="list-style-type: none"> <li>ONTAP Select</li> <li>Unity VSA</li> <li>IsilonED Edge</li> </ul>	<ul style="list-style-type: none"> <li>ASW EFS</li> <li>Azure Storage(File)</li> </ul>	
	オブジェクトベース	—	<ul style="list-style-type: none"> <li>Pure Storage Flash Blade</li> <li>EMC ECS</li> </ul>	<ul style="list-style-type: none"> <li>Scality RING</li> </ul>	<ul style="list-style-type: none"> <li>AWS S3</li> <li>Azure Storage(BLOB)</li> </ul>	



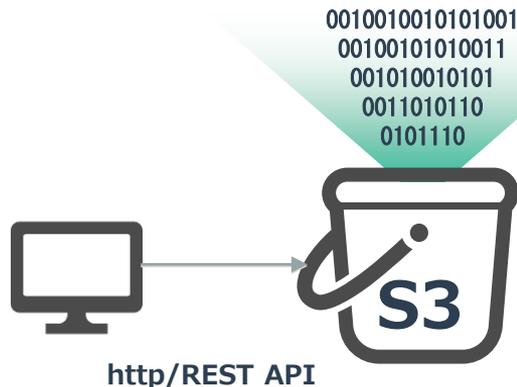
確かめること

- オブジェクトストレージはどのような用途に向くのか、セカンダリストレージ領域での使い分けを確認する

# 📁 オブジェクトストレージのおさらい

## 起源

AWS (Amazon Web Service)が2006年に投入した「S3」サービスが、オブジェクトストレージの概念を具現化した最初のサービス



## 注目される理由

### 1 ストレージ関連知識を持たないアプリケーション開発者でも利用が可能

- サーバーとの接続には、http/REST APIを利用する
- 階層型のファイルシステムを廃し、広大でフラットなファイル空間を提供

### 2 大規模、大容量を想定したアーキテクチャを採用

- 分散ファイルシステムによるスケールアウト構成の場合が多く、ノードの追加による拡張に対応しやすい
- スループット指向の性能特性
- 結果整合性(Eventually Consistent)のモデル

## 注目される理由

- 大規模・大容量を見たことに焦点を当てており、NASのような便利機能はない
- シーケンシャルファイルを格納していきような環境に向く
- ランダムアクセスは仕組み上、プロトコル上も困難



広く普及したAWS S3は、事実上の業界標準とされており、現在では、非常に多数のS3互換製品やサービスが存在し、**オンプレミスでもS3互換のオブジェクトストレージを構築することが可能。**

# 大規模ストレージにおける典型的な課題

	一般的なNASにおける課題	オブジェクトストレージでの解決
拡張性の限界	<ul style="list-style-type: none"><li>● 接続可能なディスク本数、ラックスペース、耐荷重、電力等の物理的な制約</li><li>● ファイル数、ファイルサイズ、ディレクトリサイズ、inode数等の論理的な制約</li></ul>	<ul style="list-style-type: none"><li>● ネットワークの接続性があれば拡張可能</li><li>● ファイル数、ファイルサイズ、ディレクトリサイズ、inode数等は実質無制限</li></ul>
耐障害性の限界	<ul style="list-style-type: none"><li>● RAID、ホットスベアでどこまで多重障害に耐えられるか？</li><li>● 実質選択肢がほとんどないために堅牢性については設計できない</li></ul>	<ul style="list-style-type: none"><li>● イレイジャーコーディングと自己治癒機能でRAIDを凌駕する堅牢性を実現</li></ul>
ボトルネック	<ul style="list-style-type: none"><li>● 一度ボトルネックが発生すると有効な対策が殆ど無い</li><li>● ディスクを増やしても本当に性能が向上するかはトライアンドエラー</li></ul>	<ul style="list-style-type: none"><li>● データがリバランスされる為に各サーバのリソースを最大限に有効活用</li></ul>
リプレース	<ul style="list-style-type: none"><li>● 煩雑でリスクの高いデータ移行作業、サービス切替作業</li></ul>	<ul style="list-style-type: none"><li>● 既存のシステムに新しいサーバを増設した後に古いサーバを切り離せばデータ移行や</li><li>● 切替作業を伴うことなくハードウェアのリプレースが完了</li></ul>

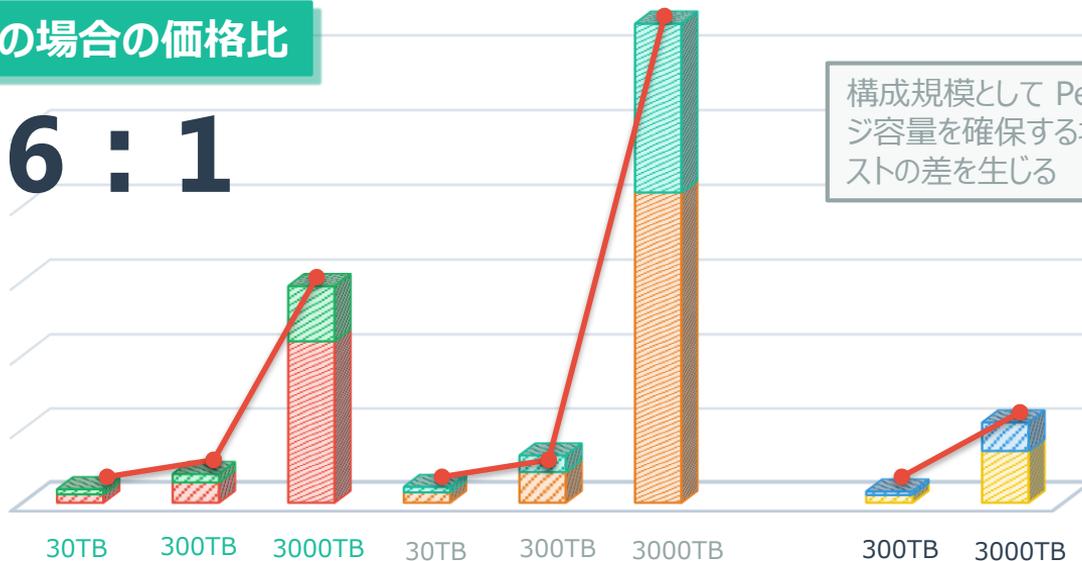
小・中規模では特に問題にならない部分でもセカンダリストレージとしての用途、特にコールドデータの扱いにおいては一般的なNASで解決できないことがある

# 価格シミュレーション

3,000TBの場合の価格比

3 : 6 : 1

構成規模として Petabyte クラスのストレージ容量を確保する場合に大きなストレージコストの差を生じる



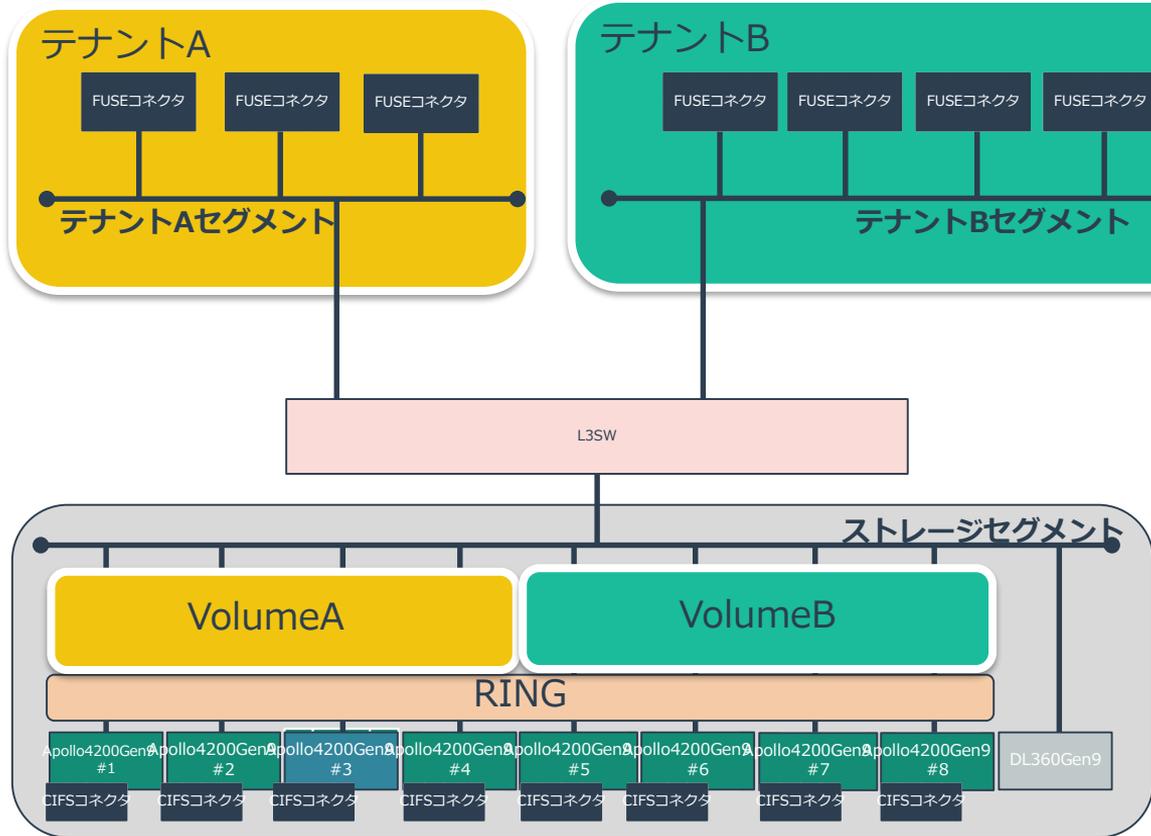
従来ストレージ

バックアップ専用  
アプライアンス

オブジェクト  
ストレージ

# 🏠 どういったパフォーマンスを示すか

検証環境構成  
(Scarity RING)



## ■ Read条件

- ファイルサイズは2MB
- ファイルアクセスは複数のファイルに対してRead
- 最大でFUSEコネクタ9台を並列して実施
- fioの多重度は512
- ブロックサイズは128KB

## ■ Write条件

- ファイルサイズは4MB
- ファイルアクセスは複数のファイルに対してWrite
- 最大でFUSEコネクタ9台を並列して実施
- fioの多重度は512
- ブロックサイズは256KB

# Scality RING性能確認試験結果

No.	種類	種類	クライアント数		目標スループット		QoS スクリプト	Arista QoS	結果スループット	
			テナントA	テナントB	テナントA	テナントB				
1	FUSE	Read	3	6	14Gbps		-	無	20.6Gbps	
2	FUSE	Write	3	6	7Gbps		-	無	7.1Gbps	
3	CIFS	Read	3	3	2Gbps		-	無	12.8Gbps	
4	CIFS	Write	3	3	3Gbps		-	無	9.6Gbps	
5	FUSE	Read	3	3	8Gbps	6Gbps	-	有	8.2Gbps	5.8Gbps
6	FUSE	Read	3	3	8Gbps	6Gbps	-	有	7.8Gbps	6.1Gbps
7	FUSE	Write	3	3	2Gbps	2Gbps	-	有	2.3Gbps	1.5Gbps
8	FUSE	Write	3	3	2Gbps	3Gbps	-	有	1.3Gbps	3.0Gbps
9	FUSE	Read	3	3	8Gbps	-	有	有	想定通り 詳細後述	
10	FUSE	Read	3	3	-	6Gbps	有	有	想定通り 詳細後述	

# オブジェクトストレージの適用範囲



## 非書き換えデータ

書き込み後に編集をすることはなく、読み込まれるだけのデータ



写真



音声



動画



ログ



## 大容量データ(かつ、非書き換え)

バックアップ領域、アーカイブ用に安価なストレージ領域が必要な場合



バックアップ



アーカイブ



## 将来予想が困難なデータ

将来の需要を予測しきれず、容量や性能の要件を定義しづらいデータ



ウェアラブル



センサー



自動車



## 頻繁に更新(書き換え)のあるデータ

頻繁にデータの更新が行われ激しいIOを伴うようなデータには向いていない



データベース



グループウェア



仮想基盤  
データストア

# 🏠 「これから」の要件への適合性

	要件（これから）	ファイル	オブジェクト
利用業務・ 必要容量の把握	使いたいときに使いたいだけ 必要な時に必要な容量を必要なだけ使いたい	✗	✗
接続性	SAN・NAS利用 + 考慮したくない	○	✗
バックアップ アーカイブ	より大容量でかつ安価	△	○
サービスレベル	より高速でかつ落ちない	○	○
災害対策	被災時のみの稼働とコスト	✗	✗

# クラウドを鑑みた使い分け

## プライマリストレージ

### SORシステム向け



Block



1

## セカンダリストレージ

### SOEシステム向け

現状は非常にサイロになっているケースが多くDXが生まれにくい



文書ファイル群

File



3



Webなどのコンテンツ群

Object



2



システムログ センサーログ

BIG-DATA



システム・APP  
バックアップ アーカイブログ  
REDOログ

Backup



ビジネスの差別化を行うに当たり  
この領域にAIを積極的に取り入れることが必要

ファイル

オブジェクト

## ③DRを考える

# 新たな選択肢の登場

## 現在での実装について

新しい選択肢が登場してきており、改めてメリット・デメリットなどの検討を行うフェーズが来ている

立地の  
考え方

従来の概念通り、自前データセンターを用意する方式は依然として有力候補であるものの、クラウドの凄まじい進化(金融機関向けの安全対策への適合やリソース提供方式の多様性、コストの適正化)により、1つの選択肢になり得る可能性が出てきている

物理的に  
対応



- 被災時の切り替わりについては多くの事例あり
- 平時のDR先設備の有効活用は更に議論が必要

クラウドで  
対応



- DR先として活用可能かは検討・検証が必要
- 平時のDR先設備の有効活用についてクラウドの特性と組み合わせることで解決できる可能性

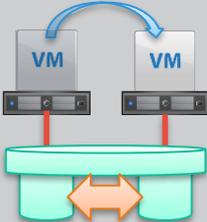
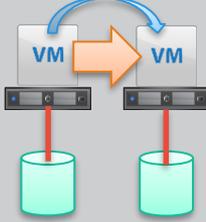
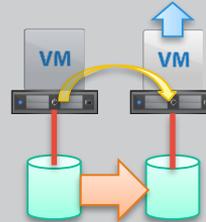
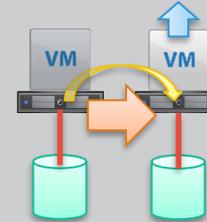
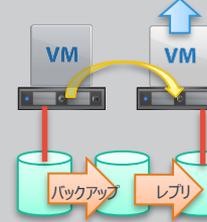
構成

DRを実施する際のデータレプリケーションにおいて、専用装置もしくはストレージ自身で実施する方式は依然として優先度の高い方式であるが、クラウドの進化により、ハードウェアでの実現だけでなく、ソフトウェアでの実現も視野に入れ検討が必要

# DR DCを考える上での設置方式

	物理的に対応 メインDC → DRDC	クラウド対応 メインDC → クラウド		
	建屋そのものを新設	既存のデータセンター業者よりラックなどの単位でレンタル		
	クラウドのIaaSやPaaSなどを活用	ホステッド型の活用		
概要	建屋を新設し、DRのDCとして整備する。インフラ設備を導入しシステムを構築する	既存DCからラック等をレンタルし、DRのDCとして整備する。インフラ設備を導入しシステムを構築する	クラウドリソースを活用し、ネイティブサービス上でシステムを立ち上げる (AWS EC2, Azure VMなど)	クラウドリソースを活用し、オンプレミス環境と同等のシステムを立ち上げる (VMware Cloudなど)
保守性	△ 原因特定まで自社にて可能	△ 原因特定まで自社にて可能	○ インフラ部分のクラウド業者に任せることが可能	○ インフラ部分のクラウド業者に任せることが可能
移行性	○ 同一アーキテクチャのため容易	○ 同一アーキテクチャのため容易	✗ そのままのディスクイメージを起動できないためツール等が必要	○ 同一アーキテクチャのため容易
機密性	○ 物理的、システムが完全自社設備のためセキュリティを担保	△ 建屋を共用しており、完全な独立ではないため	✗ 全てのコンポーネントを共用	△ DCとネットワークを共用しており、完全な独立ではないため
柔軟性	✗ 導入済みのHWのスペックを超えることや物理的な変更が手動	✗ 導入済みのHWのスペックを超えることや物理的な変更が手動	○ 必要なスペックを必要な時に必要なだけ調達可能	△ ホステッド分のHWスペック超え不可だが、増設がサービスで可能
有休機器対処	✗ 導入済みのHWの使い方を考案する必要がある	✗ 導入済みのHWの使い方を考案する必要がある	○ 必要なくなった場合はリリースコストをリダクション可能	△ タイミングによりホステッド分の台数から削減の契約が可能
導入コスト	✗ 建屋そのものが高額。更に設備費用、運用人員の配置が必要	△ ラックのレンタル費用と設備費用が必要	○ リソースを使った分の費用が必要。平時は最小限のみの利用	△ ホステッド台数分の利用料が必要。平時でも同様に発生
有事対応コスト	○ 発動はシナリオ通りに動作させ、フォールバックにおいても既設範囲で対応が可能 (追加料金などはない)	○ 発動はシナリオ通りに動作させ、フォールバックにおいても既設範囲で対応が可能 (追加料金などはない)	✗ アーキテクチャが異なるため、深い確認を要する。フォールバック時にデータをメインDCに戻す際に課金が発生	△ 発動はシナリオ通りに動作させる。フォールバック時にデータをメインDCに戻す際に課金が発生

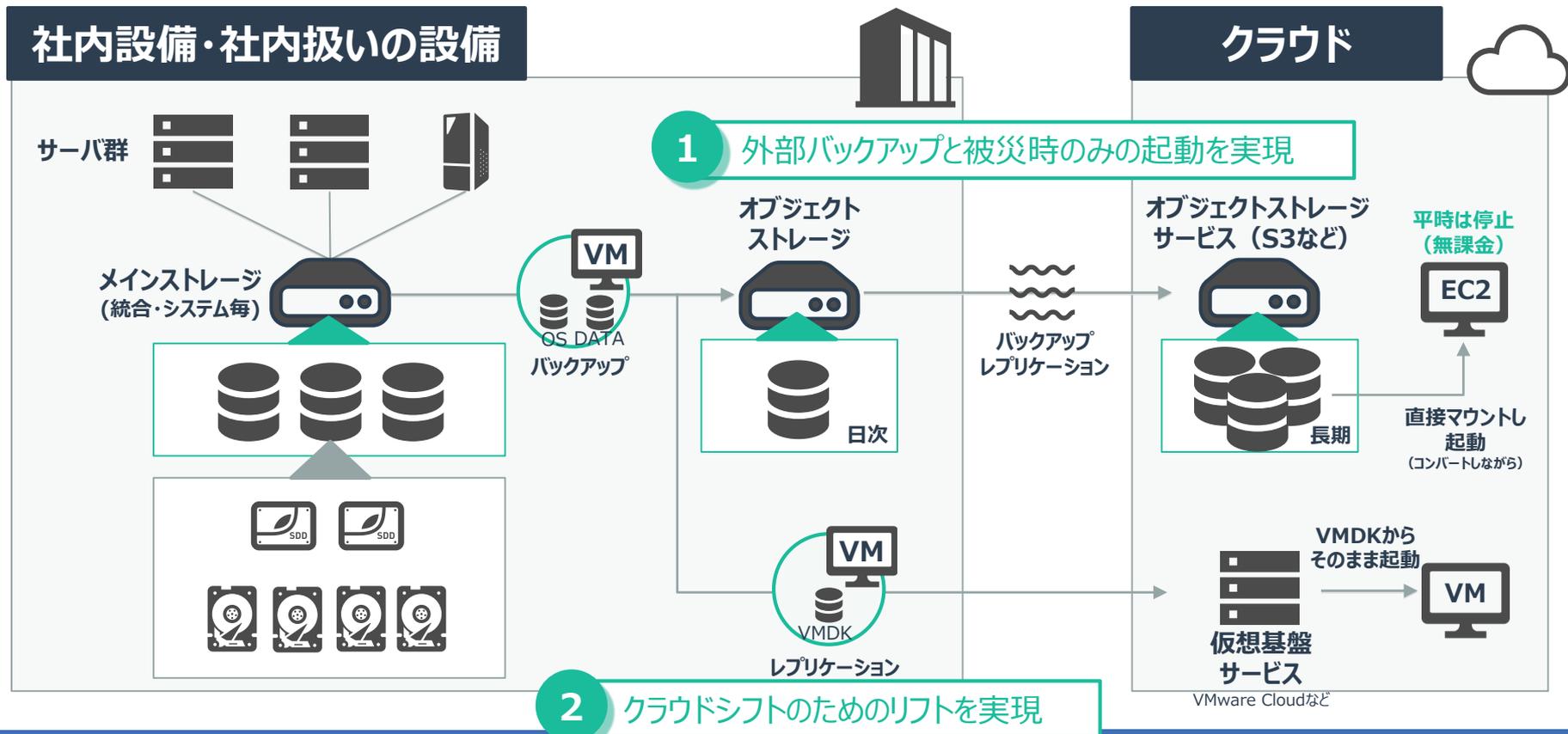
# 仮想マシン切替とデータ転送

項目 \ 方式		仮想ボリューム方式	アプリ完結方式	サーバ・ストレージ連携方式	ハイパーバイザ方式	バックアップ・リストア方式
サービスレベル	RTO	OS起動時間	数秒	数時間	数時間	数時間
	RPO	≒0	数秒から数時間	数秒から数時間	15分～48時間	1日以上
冗長方式	仮想マシンの切替方式	Active-Standby	Active-Standby	Active-WarmStan	Active-WarmStan	Active-ColdStan
	サーバ自動切替	サーバ自動切替	アプリ自動切り替え	サーバ自動切替	サーバ自動切替	サーバ自動切替
	データ転送レイヤ	ストレージ層	仮想マシン層	ストレージ層	ハイパーバイザ層	バックアップ層
	データ同期間隔	同期	非同期	同期・非同期	非同期	非同期
コスト指標		高価	非常に高価	一般的	安価	一般的
方式イメージ						
方式詳細	仮想マシン切替方式	VMwareHAによる自動切替	アプリ/DB自動切替	VMware SRMによる自動起動	VMware SRMによる自動起動	バックアップソフトによる半自動起動
	データ転送方式	拠点間仮想ボリューム	アプリ/DBの転送機能	ストレージのレプリケーション機能	ハイパーバイザのレプリケーション機能	バックアップソフトによるレプリケーション
	データ同期間隔	メイン・DR間で同期によるデータ転送	アプリ設定により短期間・長期間を選択	レプリケーション設定にて短期間・長期間を選択	ツールの自動設定。厳密な転送間隔の設定を不可	バックアップ頻度に応じてデータ転送の実施
ソリューション例		・EMC VPLEX	・アプリ作り込み ・Oracle GoldenGate	・SRDF/S,A ・VNX MirrorView/S,A	・vSphere Replicator	・Actifio ・Veeam

# 方式比較

	仮想ボリューム方式	アプリ完結方式	サーバ・ストレージ連携方式	ハイパーバイザ方式	バックアップ・リストア方式
前提となる環境	<ul style="list-style-type: none"> <li>● L2延伸がされていること</li> <li>● 回線帯域が4Gbps以上であること</li> <li>● 回線遅延が5ms以下であること</li> </ul>	<ul style="list-style-type: none"> <li>● アプリが同期するデータが遅延無く送れるWAN</li> <li>● 対応アプリケーションが選定できていること</li> </ul>	<ul style="list-style-type: none"> <li>● アプリが同期するデータが遅延無く送れるWAN</li> <li>● 仮想マシン間の依存性が整理されていること</li> <li>● IP変更しない場合はL2延伸が必要</li> </ul>	<ul style="list-style-type: none"> <li>● 仮想マシン間の依存性（起動順序など）が整理されていること</li> <li>● IP変更しない場合はL2延伸が必要</li> </ul>	<ul style="list-style-type: none"> <li>● 仮想マシン間の依存性（起動順序など）が整理されていること</li> <li>● IP変更しない場合はL2延伸が必要</li> </ul>
DC間回線への依存	高	中	中	低	低
有事切替の容易性	容易（切替なし）	容易（作り込み次第）	容易（SRMシナリオが完全であることを前提として）	容易（SRMシナリオが完全であることを前提として）	容易（オペレーションに考慮が必要なものあり）
平時での仮想マシンサイト間移動	◎	△ (移動ではなく、切り替えとなる)	× (レプリケーション対象ボリュームがDRサイトで見えない)	× (レプリケーション対象ボリュームがDRサイトで見えない)	△ (移動ではなく、複製立ち上げとなる)
BCP訓練可否	△ (片系サイトの疑似障害不可)	△ (VMの両起動が前提であるためスプリットするためには施策が必要)	△ (仕様上は可能。ただし、要件と製品仕様がフィットしないことあり)	△ (仕様上は可能。ただし、要件と製品仕様がフィットしないことあり)	△ (仕様上項可能。ただし、作り込みが必要)
RPOの遵守性	◎ (2DC間を1Volumeとして見RPO=0をシステムとして実現)	○ (アプリの設定により同期・非同期が選択でき担保可能)	○ (ストレージ側で担保可能)	× (vSphereの場合 vSpherereplicationとなり、任意のRPO指定が難しい)	△ (バックアップ頻度がRPOに直結する)
DRにおけるRAWボリュームの取り扱い	RAW、VMFSの区別なく仮想化されているため、DR切替が発生しても同一ボリュームとして取り扱いが可能	アプリベースでのレプリケーションであるため、OS/HDDとして認識していればRAWでも問題なく取り扱い可能	基本的にはサポートされているがVMFSに比べ制限事項が多い。対象システム毎に検証等が必要	不可	OS上のバックアップエージェントと連携して可能
クラウドへの展開	× (クラウドに対応機器の設置が困難)	○ (アプリベースであるため、クラウド上で同一の仕組みを構築可能)	△ (DR先にvSphereが設置できれば可能(ホステッド型))	△ (DR先にvSphereが設置できれば可能(ホステッド型))	○ (対応製品を選定すればクラウドネイティブ環境に展開可能)

# 視点毎の最良構成



# まとめ

プライマリ、セカンダリストレージの分けととも、要件の優先度に応じて**選択幅が広がったこと**

クラウドもオンプレも機能・性能については近接してきており、**技術目線**ではどちらを使用しても要件を満たせるシステムが構築可能

単純なオンプレの置き換えで運用するとオンプレを超えるコストが発生する可能性がある（従量課金システムのデメリット）**用法・用量を守って正しく使うことが肝要**

違うのは進化のクラウドは進化のスピードが速く、そのスピードについて行ければ**ビジネス大きなチャンスが生まれる**

ネットワンシステムズはネットワークベンダのイメージが強いですが、しっかりとストレージをはじめとした**プラットフォームビジネスを展開**しております。

つなぐ ∟ むすぶ ∟ かわる



net one