ストレージの基礎

Yukinori Sakashita

Z Lab

注意事項

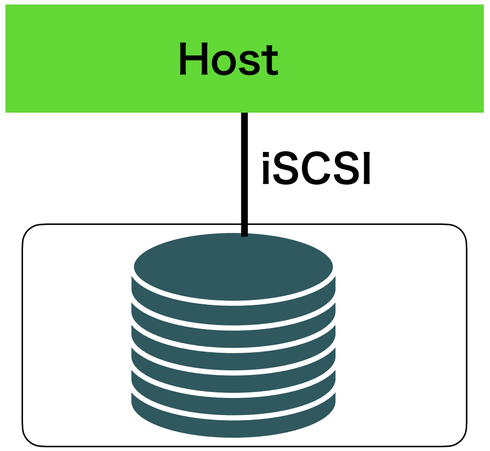
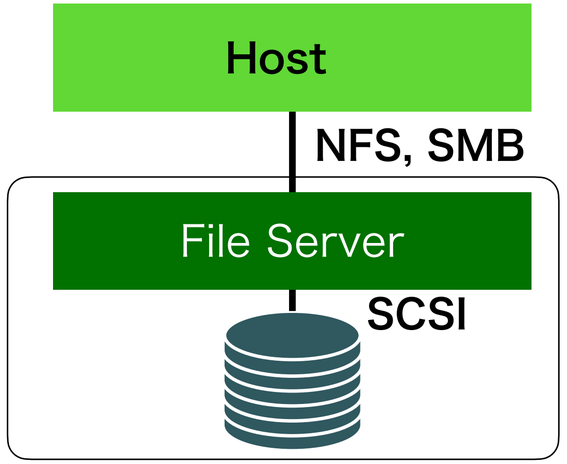
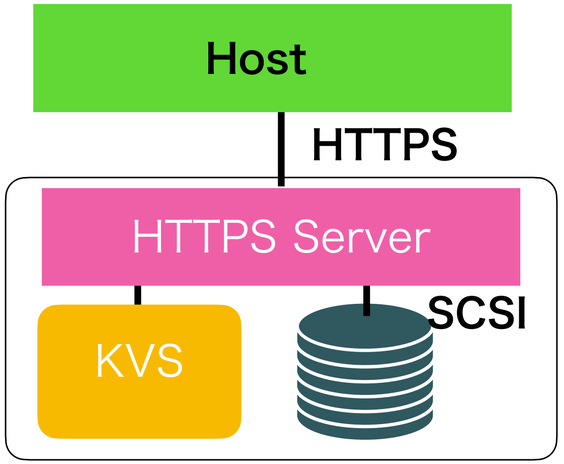
- ストレージはベンダや製品ごとでコンポーネントなどの名称が異なることがあります
- 本資料ではストレージ業界団体SNIAが定めるストレージ管理I/Fの国際標準仕様SMI-S(ISO/IEC 24775-1～8, ANSI INCITS 388-2004)、Swordfishで定めれたコンポーネントの名称およびSNIAの教育資料で使用される用語を使います

https://www.snia.org/tech_activities/standards/curr_standards/smi
https://www.snia.org/tech_activities/standards/curr_standards/swordfish

Agenda

- ストレージの種類
- ブロックストレージ
- ファイルストレージ
- ストレージの雑学

ストレージの種類

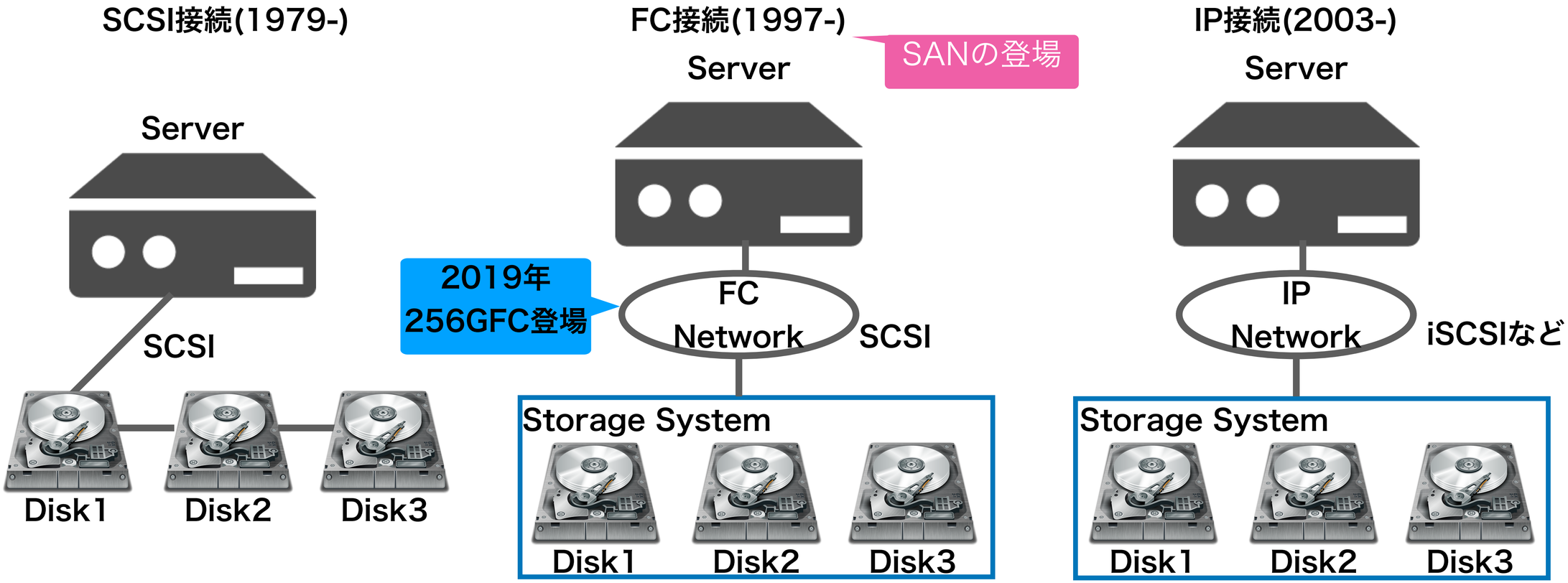
	ブロックストレージ	ファイルストレージ	オブジェクトストレージ
アーキテクチャ			
特徴	• 内蔵ドライブと同じRawデバイス • ファイルシステムは自由に選択	• ネットワークドライブ • ファイルシステムはストレージ任せ、変更不可	• オブジェクト単位でアクセス • ファイルシステムに依存しないため、大量データを格納可能
データ転送プロトコル	iSCSI, FC, NVMe	NFS,SMB	HTTPS(HTTP)
マウントパスの例	/dev/sda	\\192.168.0.1\share\hoge	https://aaa.org/storage/hoge
性能*	High	Middle	Low
主な用途	DB, OS(Boot Disk)	ファイルの共有	写真、動画格納

*一般的な傾向であり製品により変わります

ブロックストレージ

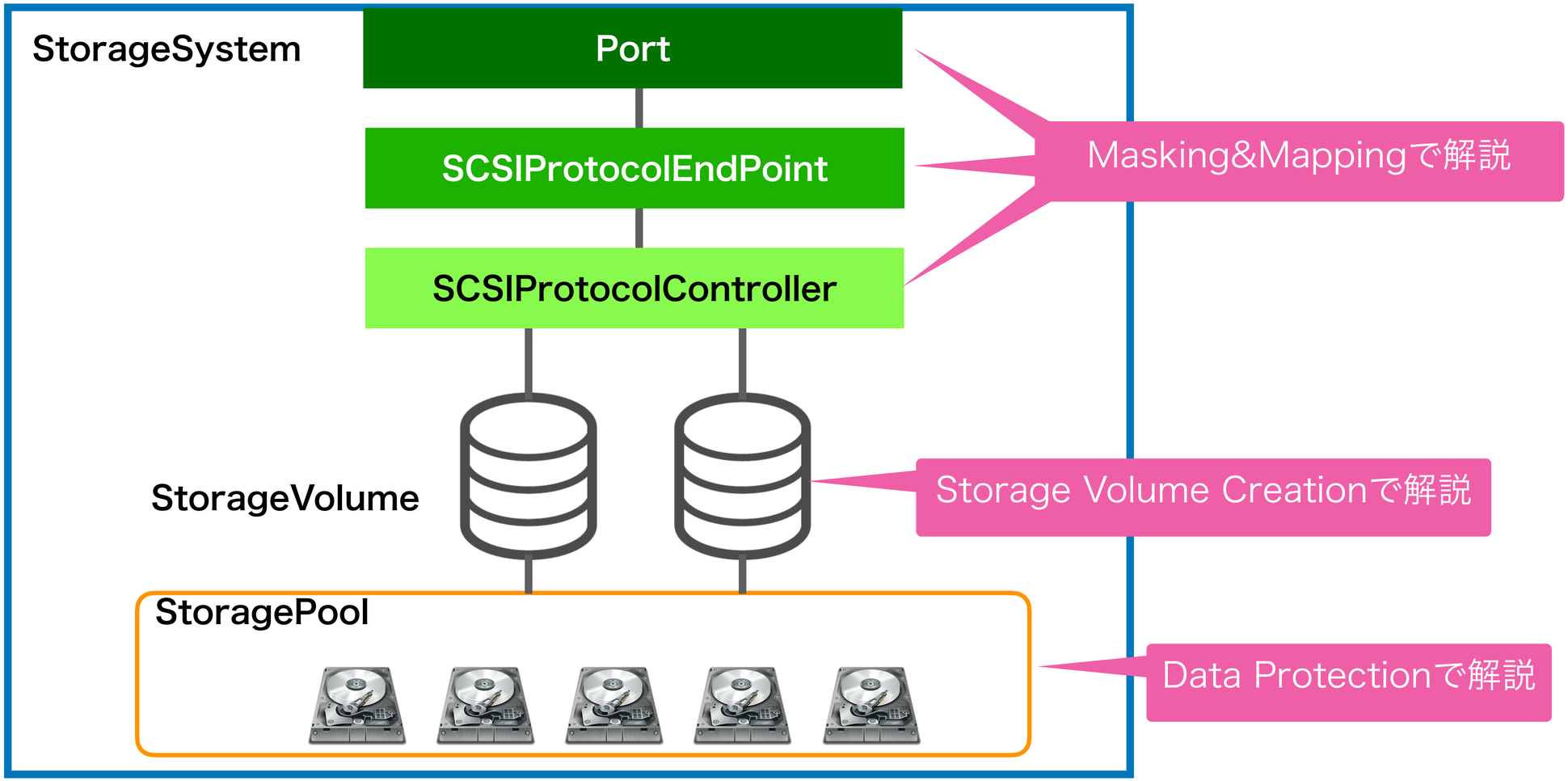
ブロックストレージと歴史

- ハードウェアのSCSIはほぼ消滅
- プロトコルとしてのSCSIは接続形態が変わっても利用



SCSI: Small Computer System Interface, SAN: Storage Area Network, FC: Fiber Channel

基本構成


StorageSystem
Port
SCSIProtocolEndPoint
SCSIProtocolController
StorageVolume
StoragePool
Masking&Mappingで解説
Storage Volume Creationで解説
Data Protectionで解説

Data Protection

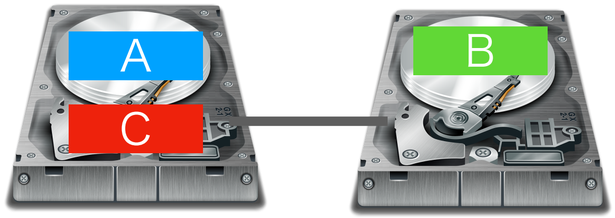
Data Protectionとは

- 複数個の物理メディア(HDDやSSD)をまとめる技術
- なぜ、複数個の物理メディアをまとめるのか？
  - 単体の物理メディア以上のサイズが欲しい
  - 1つの物理メディアが壊れてもデータが消えないようにしたい
  - 複数の物理メディアにデータを分散させ性能を向上
- どうやって実現しているのか
  - RAID
  - Replication

RAID

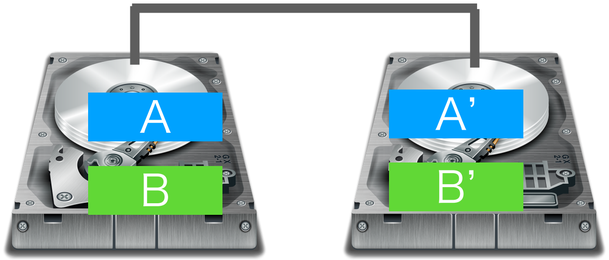
RAID 0

ストライピングにより巨大な容量の
ディスクを実現 (例. 容量20G)



RAID 1

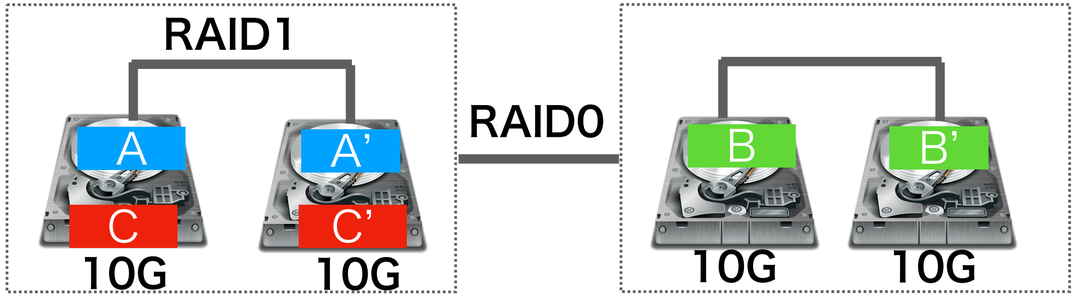
ミラーリングによりデータ保護された
ディスクを実現 (例. 容量10G)



RAID 1+0

逆のRAID0+1
もある

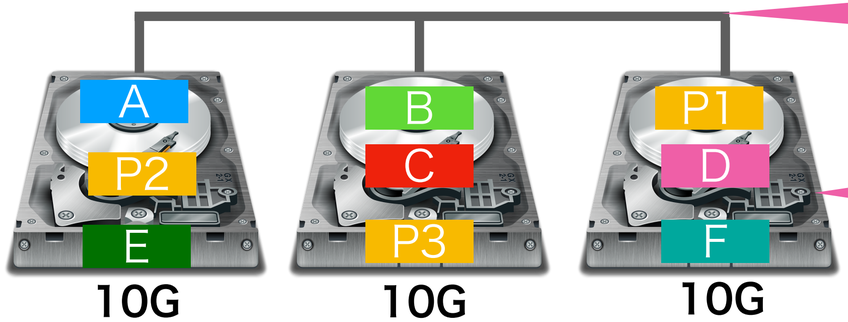
ミラーリング+ストライピングにより
データ保護+巨大な容量のディスクを実現 (例. 容量20G)



RAID 5

パリティデータにより無駄を少なくデータ保護された
巨大なディスクを実現 (例. 容量30G未満)

分散IOが可能



RAID6は
ダブルパリティ

パリティデータとは

- パリティ計算の基本はXOR(排他的論理和, eXclusive OR)

XOR

0 XOR 0 = 0
0 XOR 1 = 1
1 XOR 0 = 1
1 XOR 1 = 0

	A	B	P1
データ格納時	0000 0101	1010 1010	1010 1111
データBのディスクが障害	0000 0101		1010 1111
データBを復元 (A XOR P1)	0000 0101	1010 1010	1010 1111

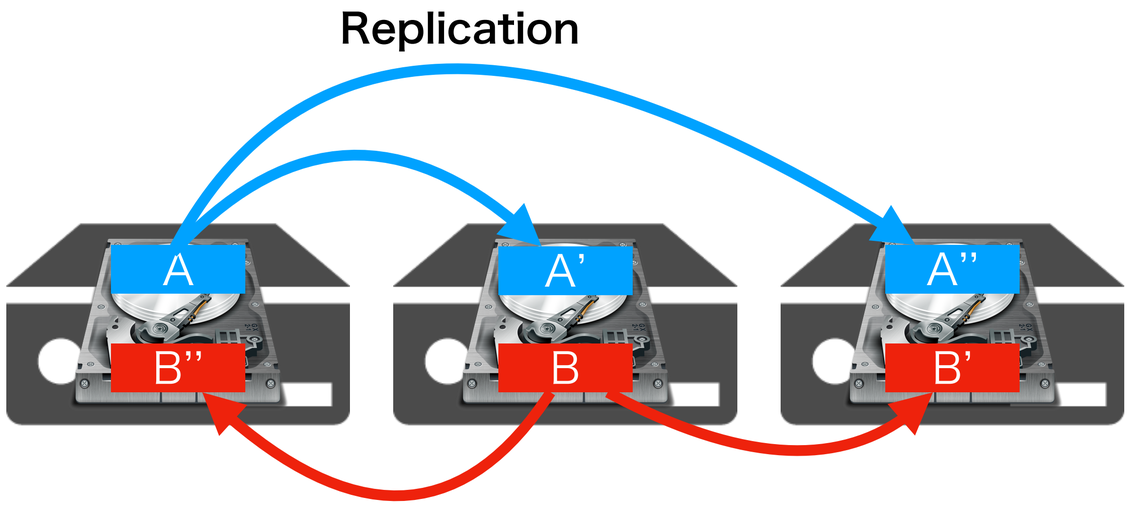
Replication

- Replicationは、データ単位でのReplicationとStorageVolume単位でのReplicationがある
- ここではデータ単位でのReplicationを解説
- Replicationの考えは基本はRAIDと同じ
- Software Defined Storageが登場し、複数サーバ(内蔵ディスク)を物理メディアとみたてData Protectionを実現
- Cephなどで採用されているTriple Replicationが有名
- 格納データを任意のサイズで分割＋パリティデータを使いデータ保護するErasure Codingもあり(RAID5, 6もErasure Codingの一つ)

Triple Replication/Erasure Coding

Triple Replication

データが書き込まれた際、別サーバに
データをReplication .
3つに分散する場合がTriple Replication



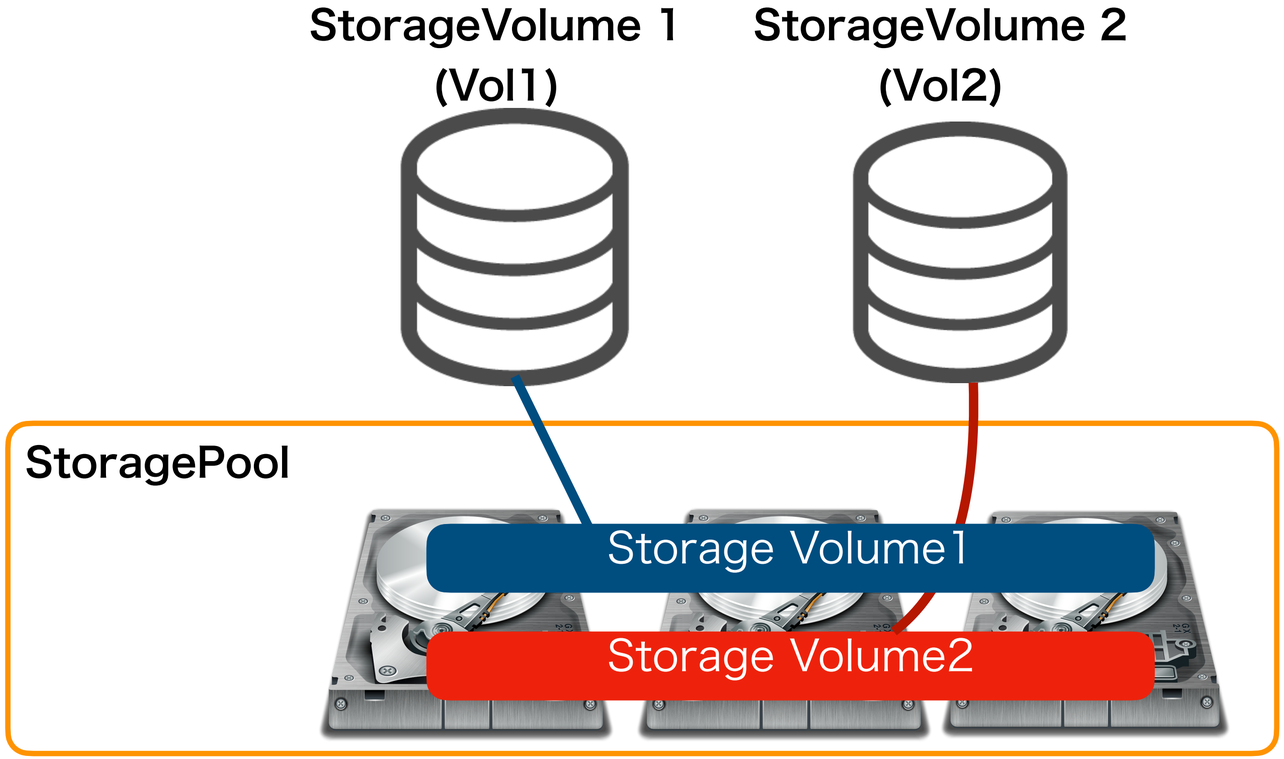
Erasure Coding

データAを分割し、複数サーバに分散配置した後
パリティデータを生成し格納

Storage Volume Creation

StoragePoolとStorageVolume

- 複数物理メディアをData ProtectionによりまとめたstoragePoolから論理デバイスとして生成したものがStorageVolume
- 論理-物理のLBAのマッピングテーブルによって実現



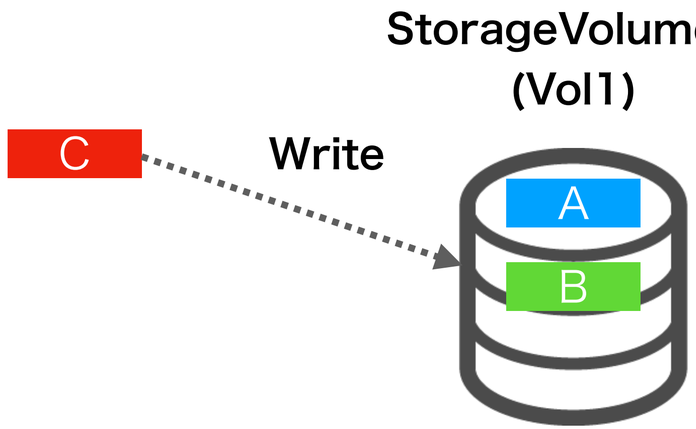
#	VolのLBA	物理メディアのLBA
Vol1	0000-5FFF	Disk1: 0000-5FFFF
Vol1	6000-AFFF	Disk2: 0000-5FFFF
Vo1	B000-FFFF	Disk3: 0000-5FFFF
Vol2	0000-5FFF	Disk1: B000-FFFFF
Vol2	6000-AFFF	Disk2: 6000-AFFFF
Vol2	B000-FFFF	Disk3: B000-FFFFF

LBAのマッピングテーブルは各社実装が異なります

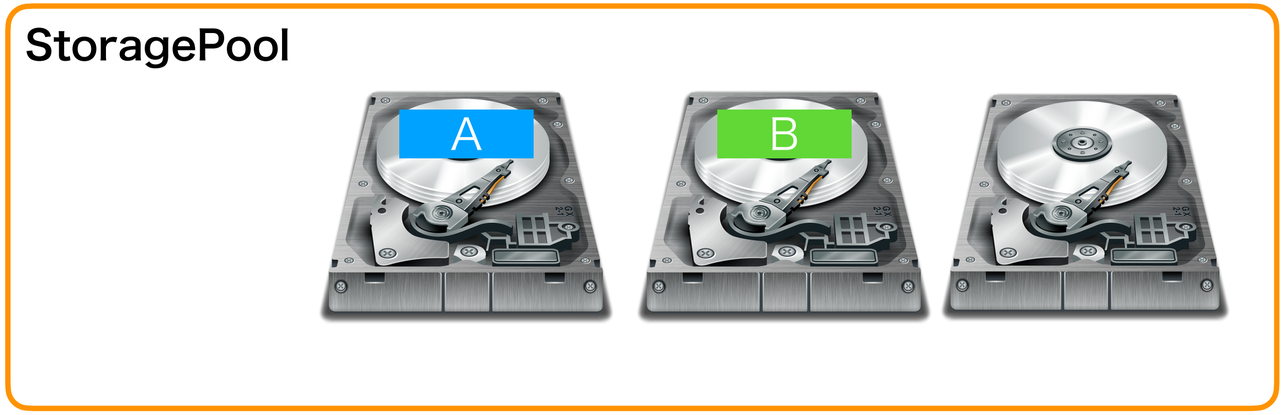
Thin Provisioning

- StorageVolumeには大きく2つのタイプあり
- Thick Provisioning
  StorageVolumeの作成時に全てのサイズ分の物理メディアの領域を
  確保する方式
  - Read/Write高速、容量効率低い
- Thin Provisioning
  StorageVolumeの作成時は物理メディアの領域を確保せず、データが書き込
  まれる都度、物理メディアの領域を確保する方式
  - Read高速, Write低速(物理メディアの割り当て発生時)、容量効率高い
  - サーバにはThick Provisioningと同様のサイズで見える

Thin Provisioning



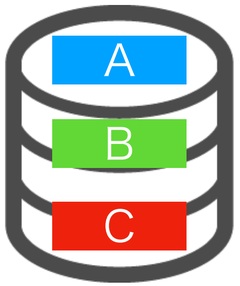
#		VolのLBA	物理メディアのLBA
Vol1	Data A	0000-0100	Disk1: 0000-0100
Vol1	Data B	1000-1100	Disk2: 0000-0100



LBAのマッピングテーブルは各社実装が異なります

Thin Provisioning

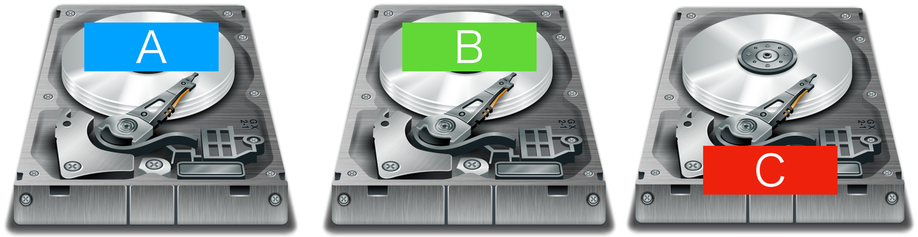
StorageVolume 1
(Vol1)



#		VolのLBA	物理メディアのLBA
Vol1	Data A	0000-0100	Disk1: 0000-0100
Vol1	Data B	1000-1100	Disk2: 0000-0100
Vol1	Data C	1200-1300	Disk3: 1000-1100

Thick Provisioningに比べ論理-物理の
LBAマッピングテーブルが巨大になる。
多段のマッピングテーブルの実装ベンダもあり

StoragePool

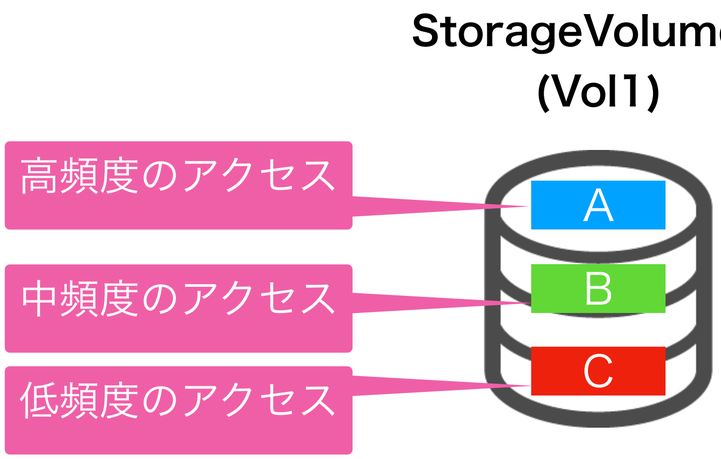


LBAのマッピングテーブルは各社実装が異なります

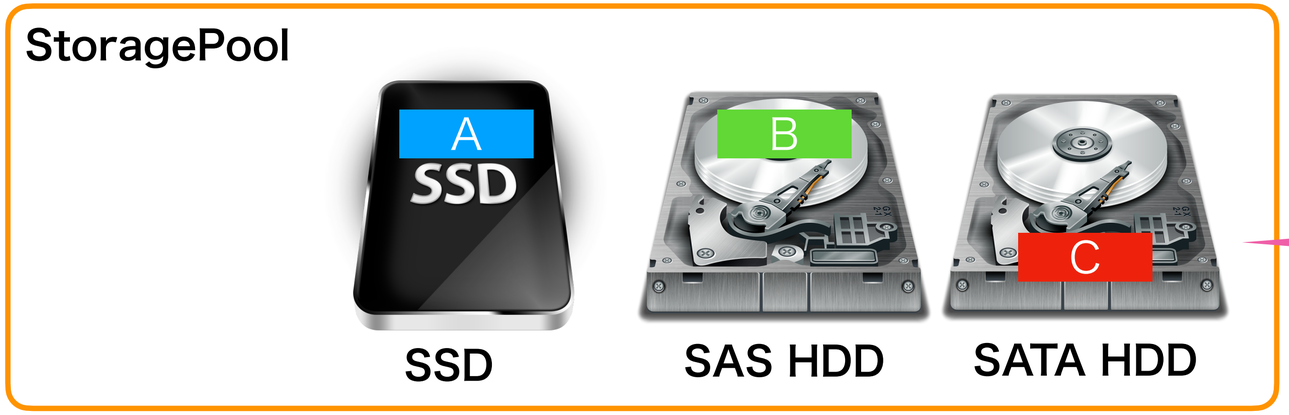
Automated Storage Tiering

- 複数の種類のHDDやSSDの登場により、物理メディアに性能・コスト差が生じ始めた(2008年頃)
  - SSD: 高速, 高価格
  - SAS HDD: 中速, 中価格
  - SATA HDD: 低速, 低価格
- 高頻度にアクセスするデータは高速/高価なメディア、中・低頻度のアクセスのデータは中・低速/中・低価格に自動でデータの配置場所を移動させ性能とコストの最適化を図るAutomated Storage Tieringが登場

Automated Storage Tiering



#		VolのLBA	物理メディアのLBA
Vol1	Data A	0000-0100	Disk1: 0000-0100
Vol1	Data B	1000-1100	Disk2: 0000-0100
Vol1	Data C	1200-1300	Disk3: 1000-1100

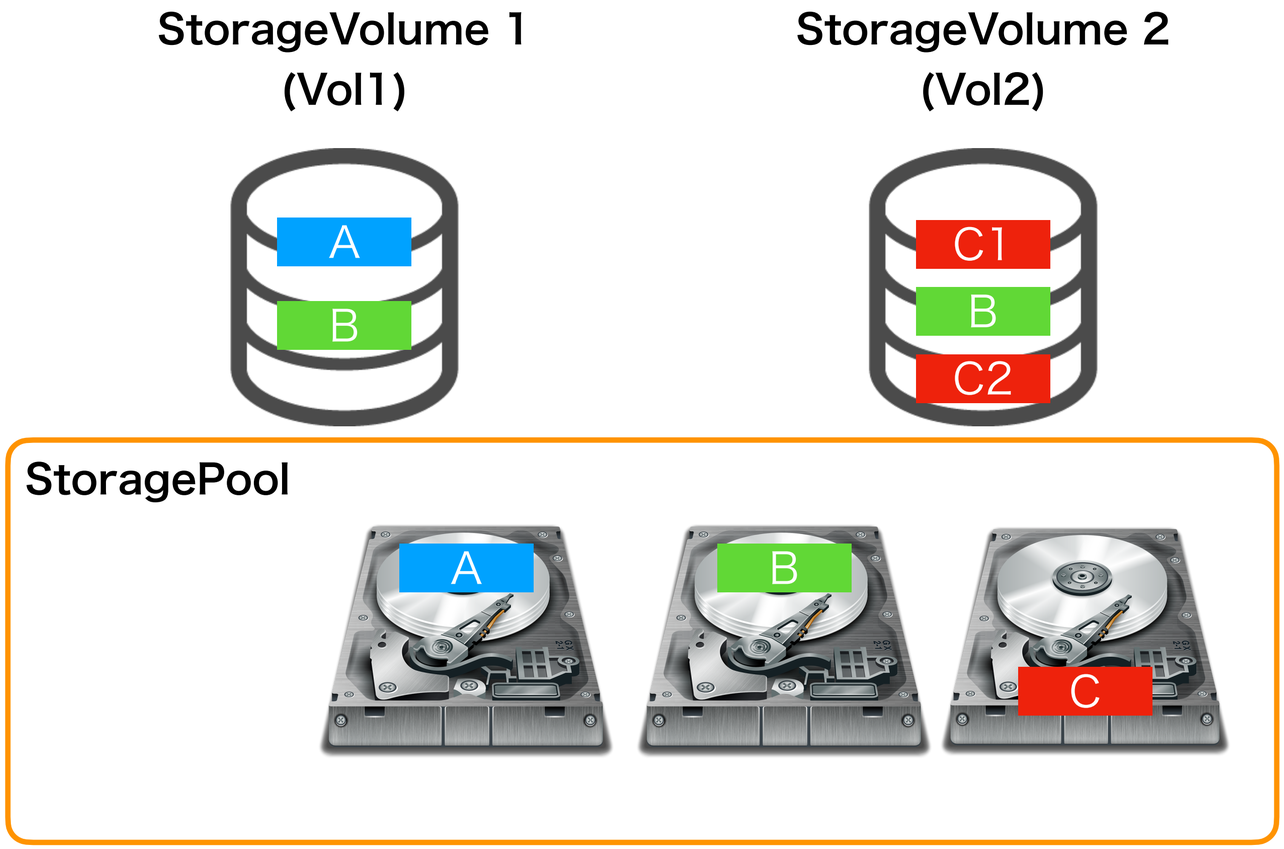


アクセス頻度を定期的に監視。
アクセス頻度によってはデータを
別メディアへ移動

LBAのマッピングテーブルは各社実装が異なります

Deduplication (重複排除)

- 物理メディアの容量削減のため、重複データは保存せずポインタ参照



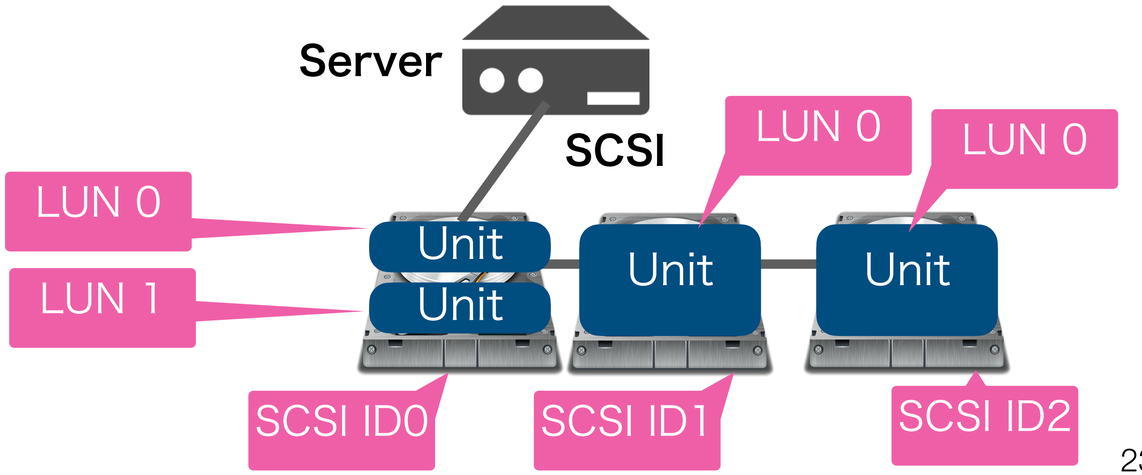
#		VolのLBA	物理メディアのLBA
Vol1	Data A	0000-0100	Disk1: 0000-0100
Vol1	Data B	1000-1100	Disk2: 0000-0100
Vol2	Data C1	1200-1300	Disk3: 1000-1100
Vol2	Data B	&Vol1.DataB	null
Vol2	Data C2	&Vol2.DataC1	null

LBAのマッピングテーブルは各社実装が異なります

Masking&Mapping

サーバとストレージの接続 (SCSI)

- SCSIバスではシリアル/パラレルインターフェースにて数珠つなぎ
- SCSI IDとLUNによってユニットを識別
  - 各DiskにはSCSI ID (0, 1, 2…)をユニーク番号として設定
  - 各Diskの論理的なユニット(StorageVolumeに相当)に LUN(Logical Unit Number, 0,1,2…)を設定



Serverでのデバイスの見え方(SVR4系UNIX)

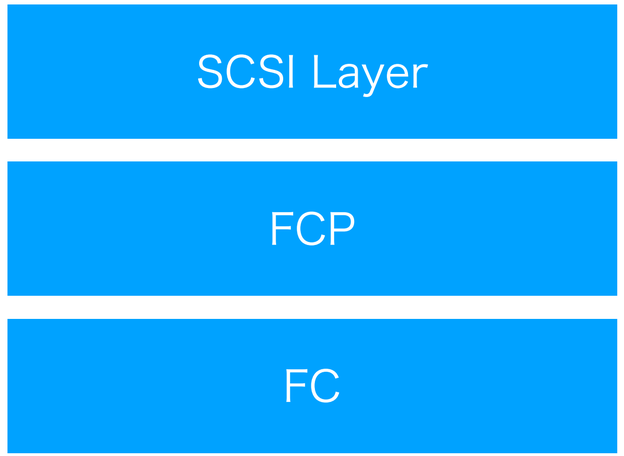
/dev/c0t0d0s0 (SCSI ID 0, LUN 0)
/dev/c0t0d0s1 (SCSI ID 0, LUN 1)
/dev/c0t1d0s0 (SCSI ID 1, LUN 0)
/dev/c0t2d0s0 (SCSI ID 2, LUN 0)

当時はSCSIデバイス=Diskのためd0固定

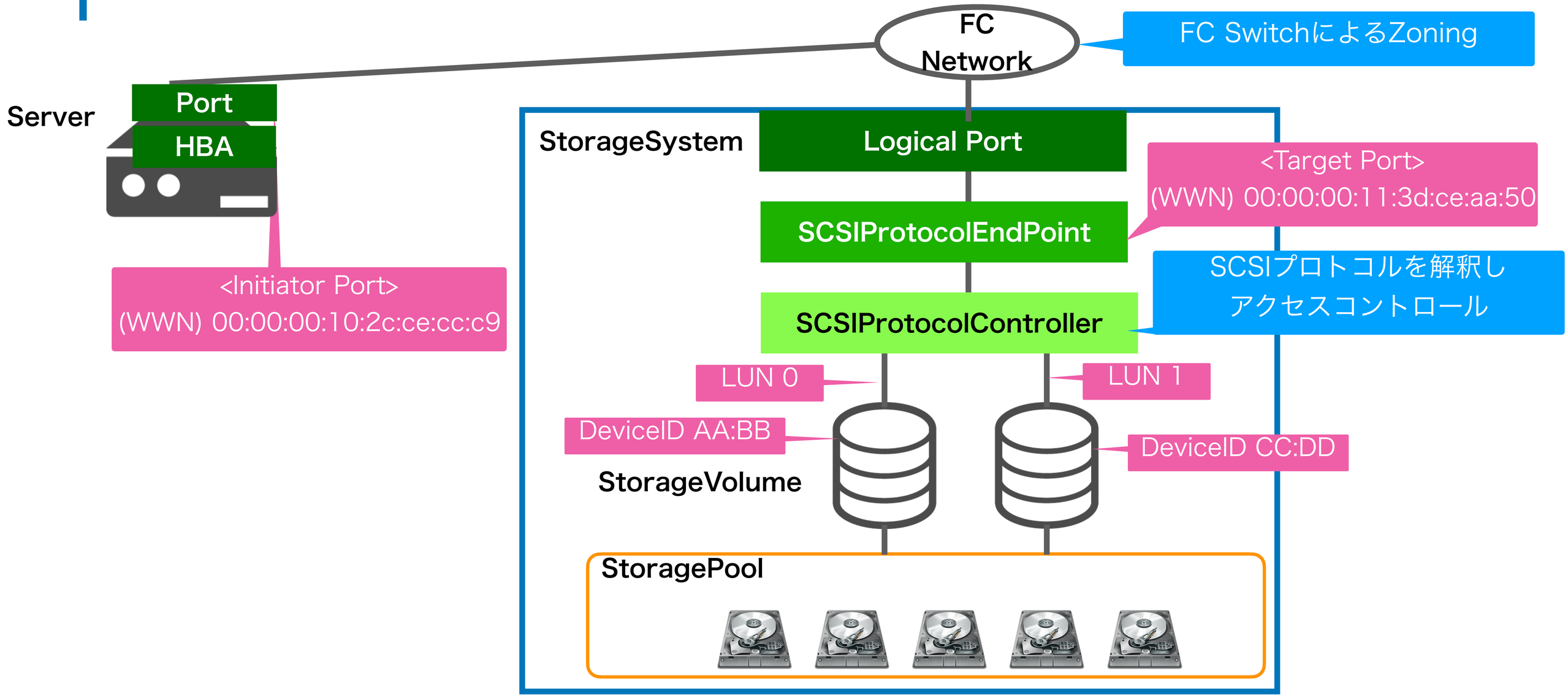
サーバとストレージの接続 (FC)

- FC接続になりネットワークにてストレージが接続。プロトコルはSCSIのまま
- WWN, LUN, DeviceIDによってStorageVolumeを識別(基本形)
  - SCSI IDが使えなくなりWWNが登場。ポート毎にユニーク値のWWNを付与
  - サーバ側かストレージ側のPortなのかの役割を持たせる必要が発生
    - サーバ側: Initiator Port
    - ストレージ側: Target Port

FCのプロトコルスタック



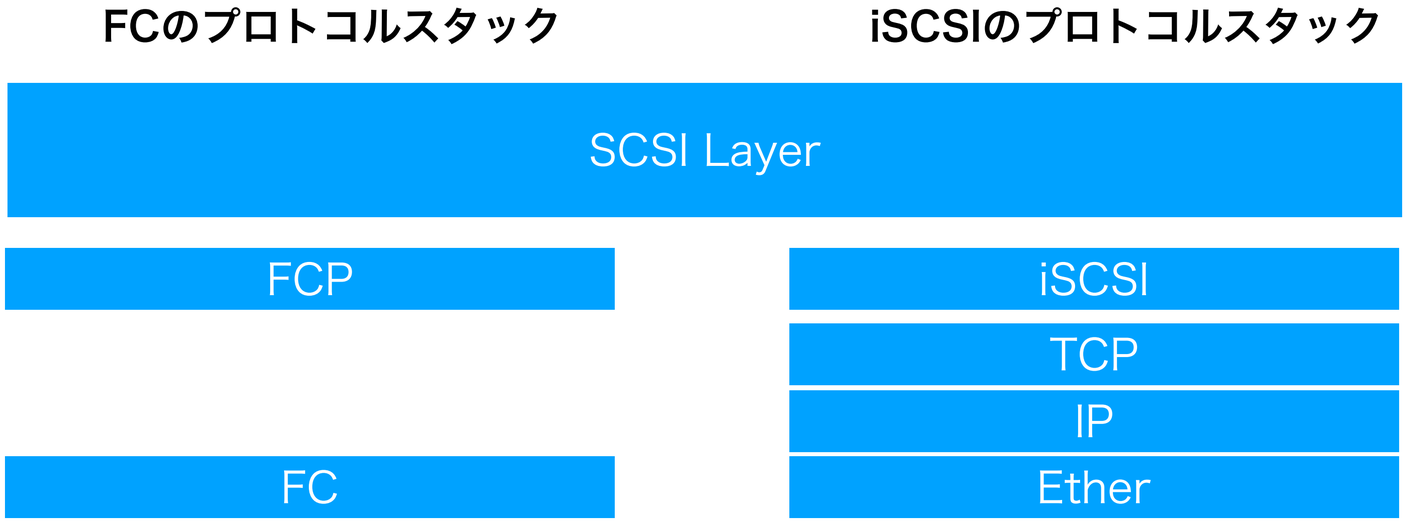
サーバとストレージの接続 (FC)



HBA: Host Bus Adapter

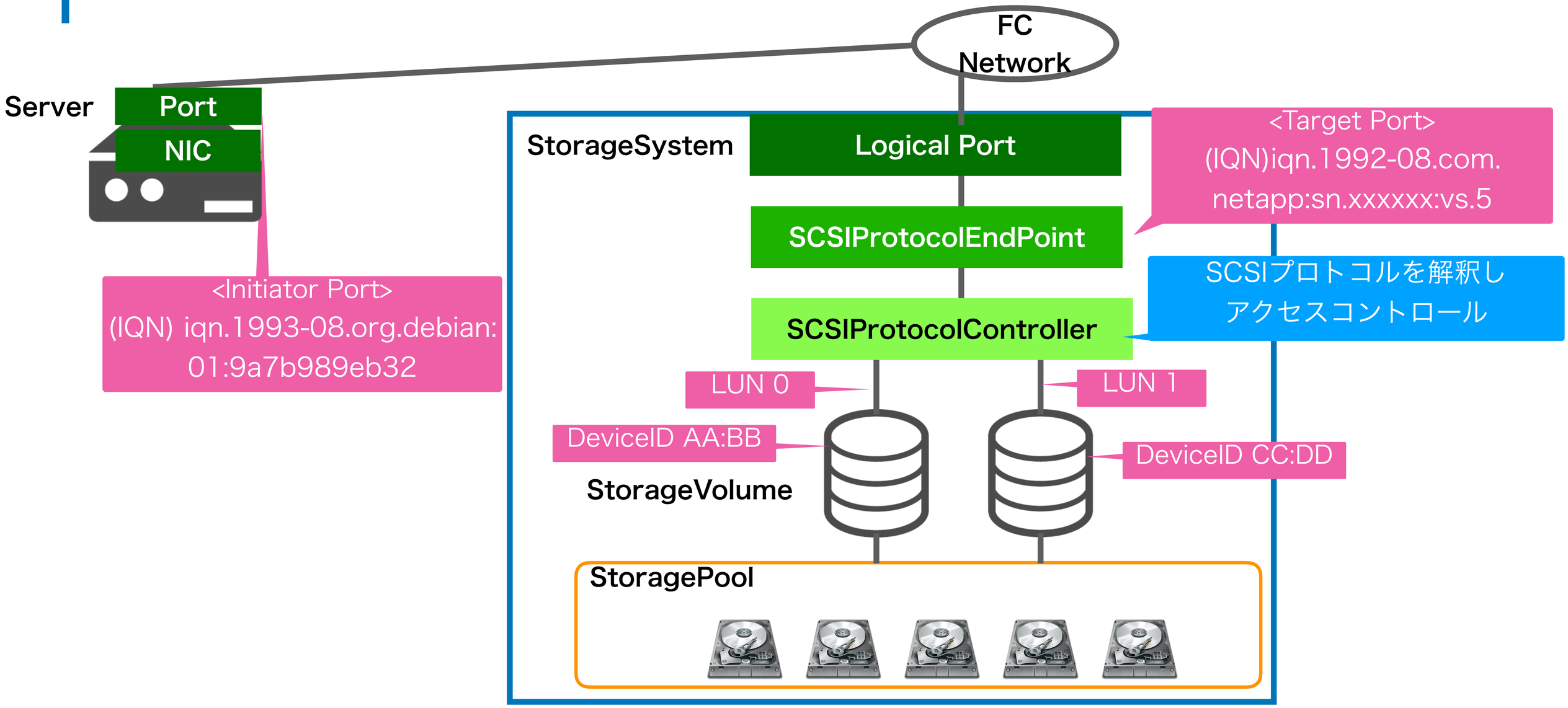
サーバとストレージの接続 (iSCSI)

- FCをIPに置き換え。アプリからみたプロトコルはSCSIのまま
- IQN, LUN, DeviceIDによってStorageVolumeを識別(基本形)
  - FCではHBAのPortにWWNがハード的にプリセット
  - iSCSIではWWNはなく代替のものが必要→IQNの登場



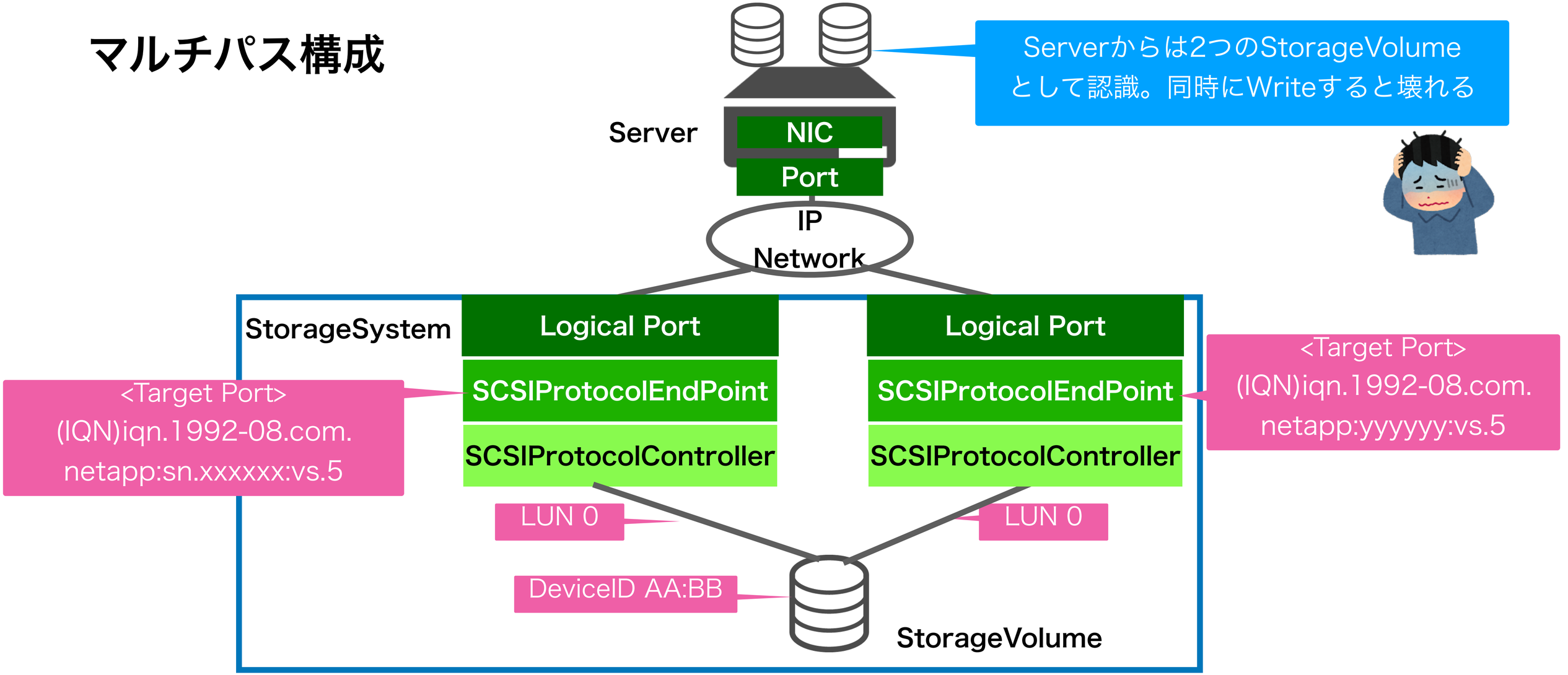
IQN: iSCSI Qualified Name

サーバとストレージの接続 (iSCSI)



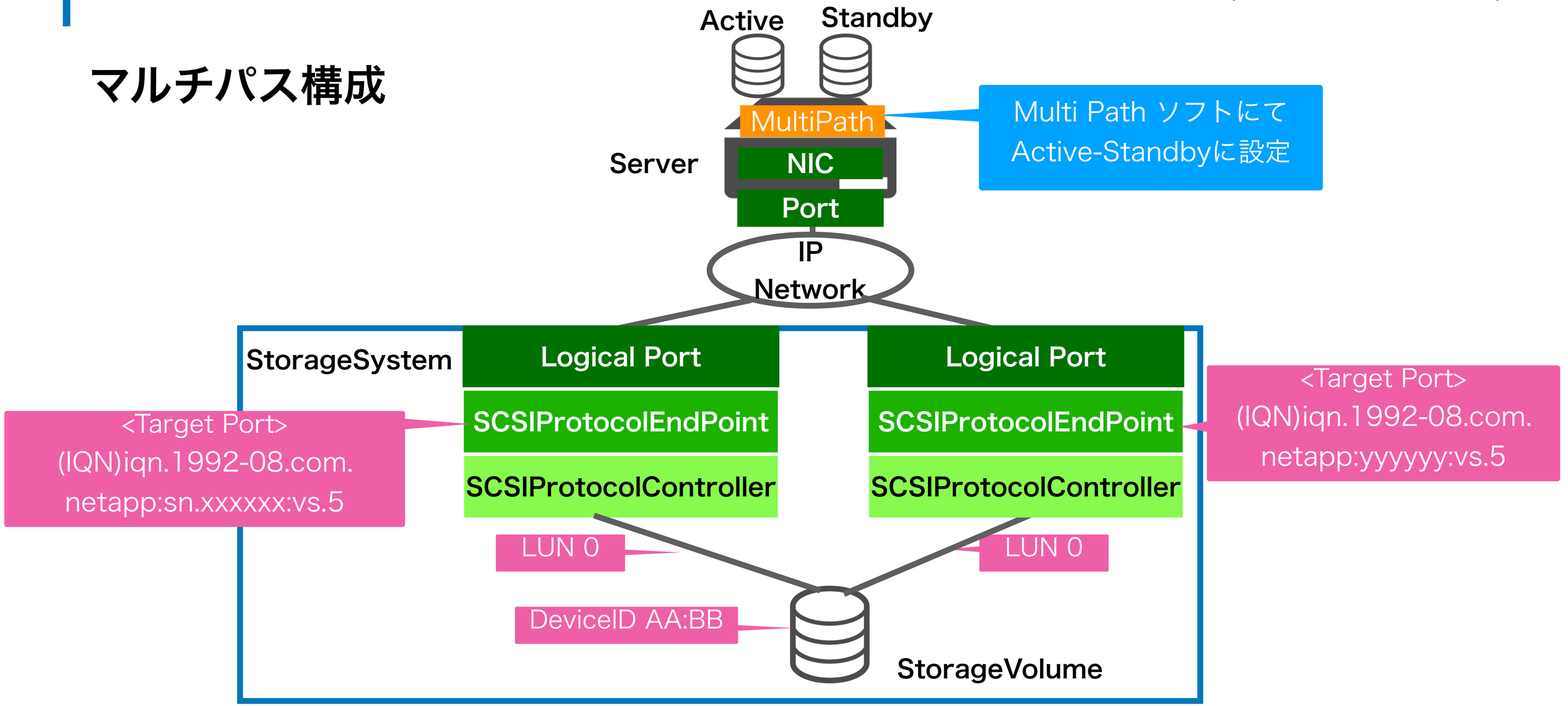
サーバとストレージの接続(応用1)

マルチパス構成



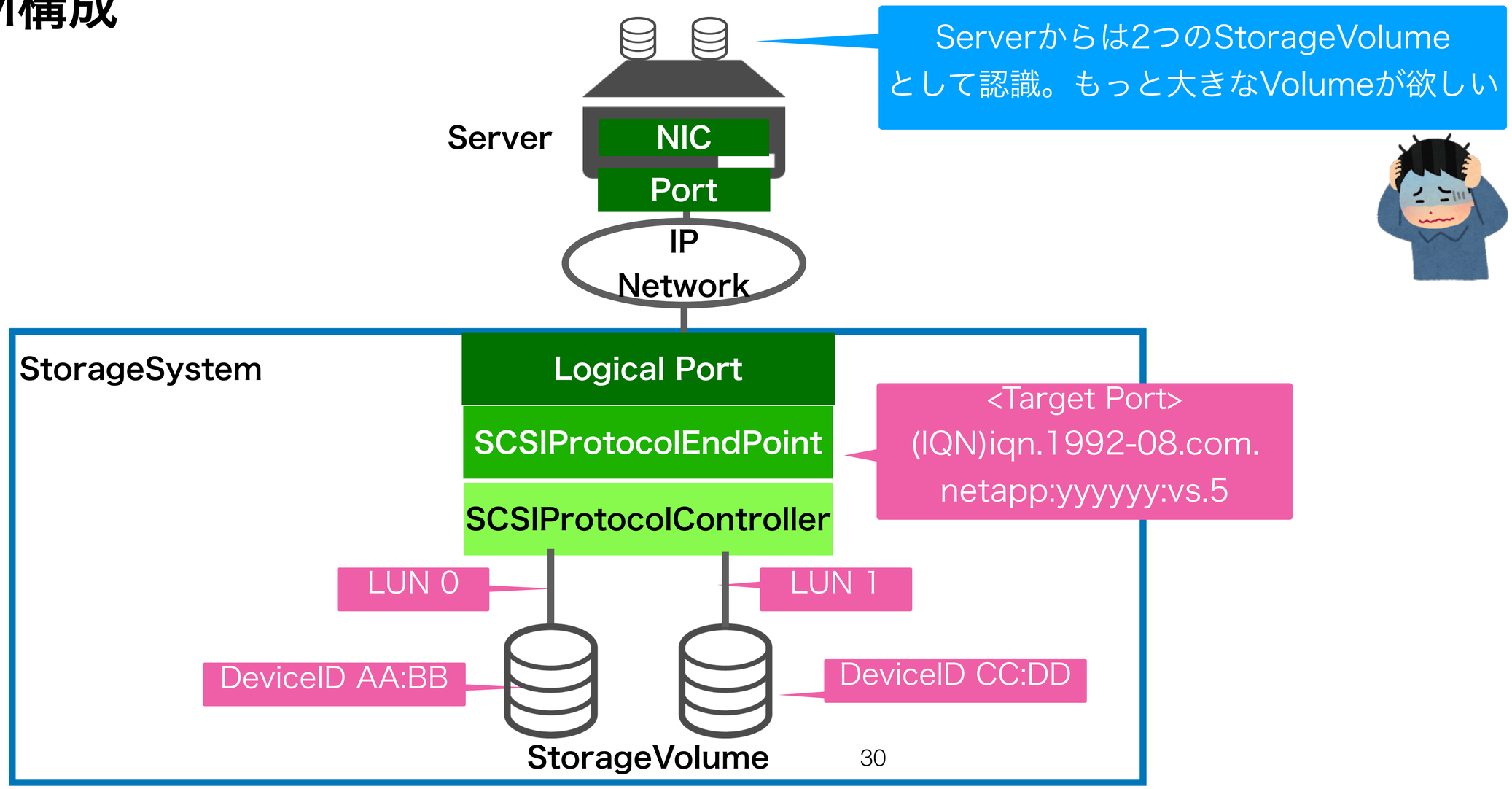
サーバとストレージの接続(応用1)

マルチパス構成



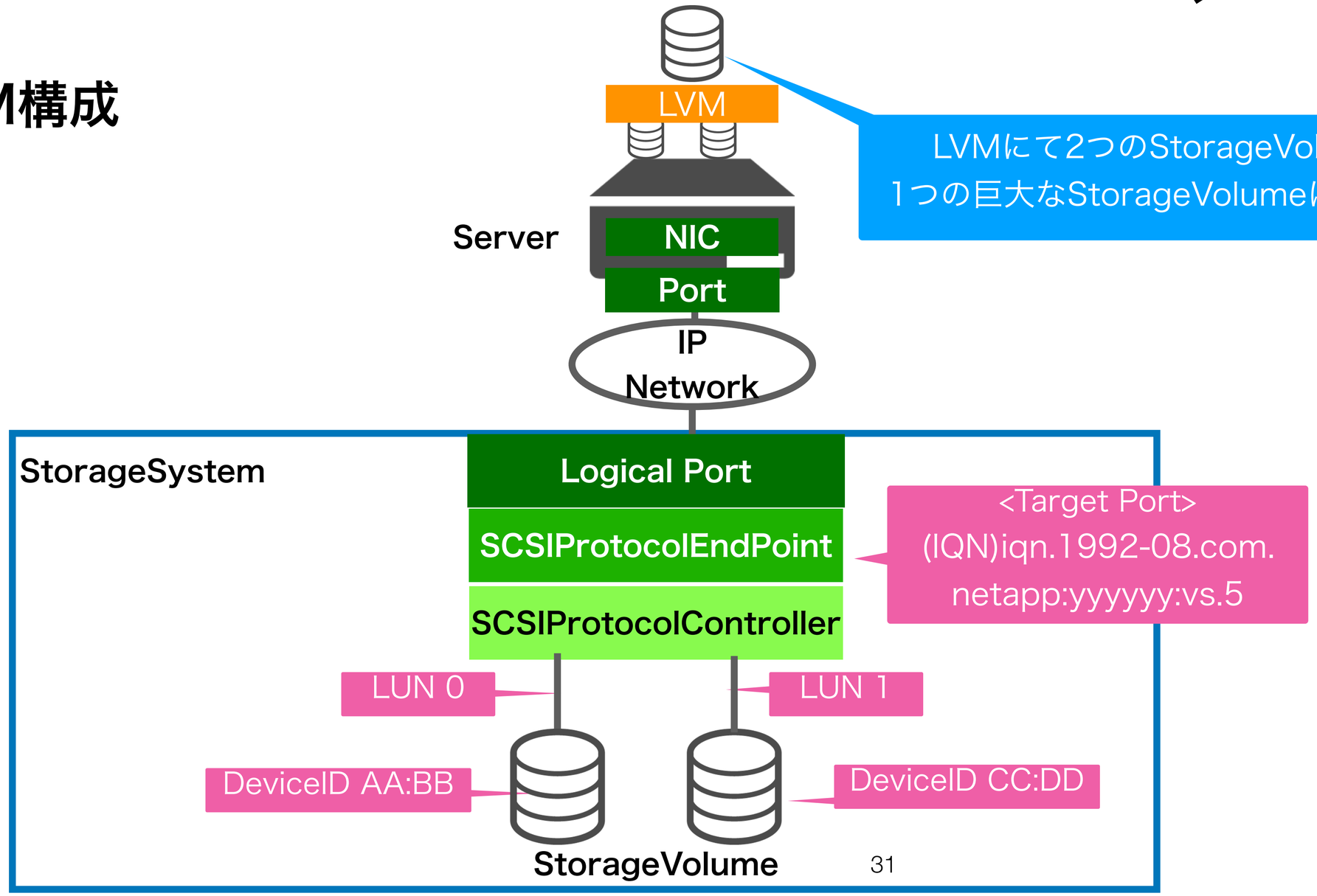
サーバとストレージの接続(応用2)

LVM構成



サーバとストレージの接続(応用2)

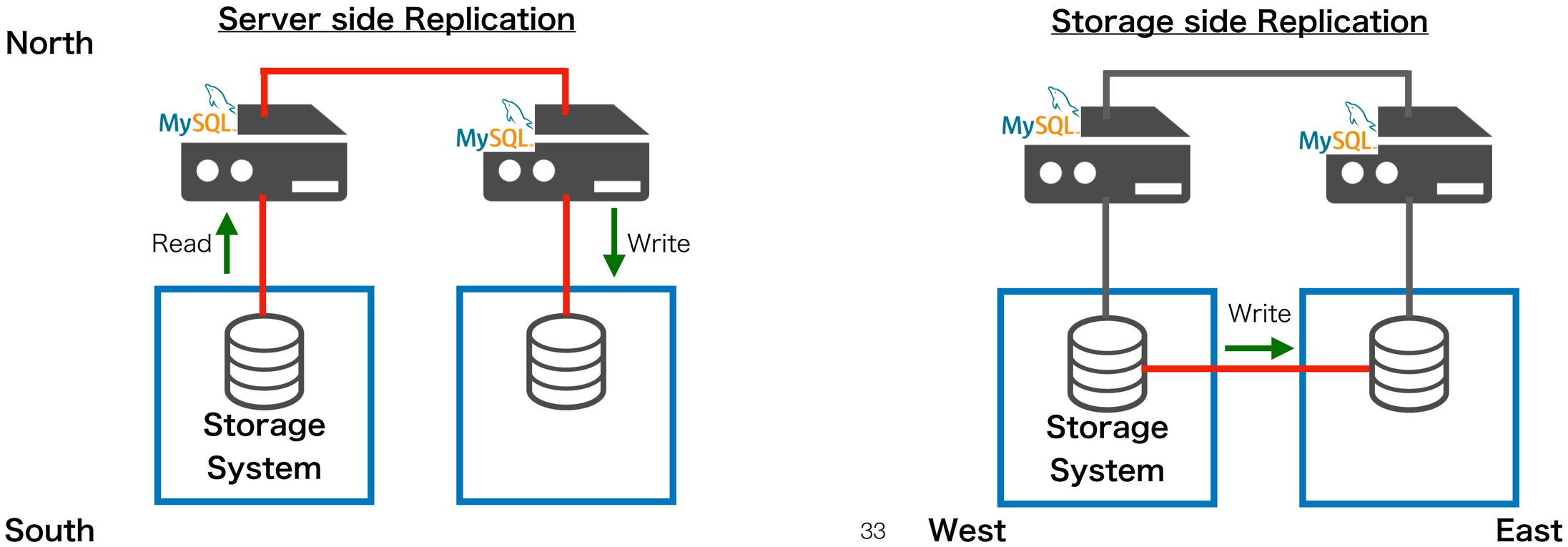
LVM構成



Replication

ストレージのReplicationの特長

- Replication処理をサーバのCPU/Memoryを使わず実行 (アプリの性能影響なし)
- ストレージ間ネットワーク(West-East)の利用により
  サーバ/ストレージ間ネットワーク(North-South)のトラフィック負荷を軽減



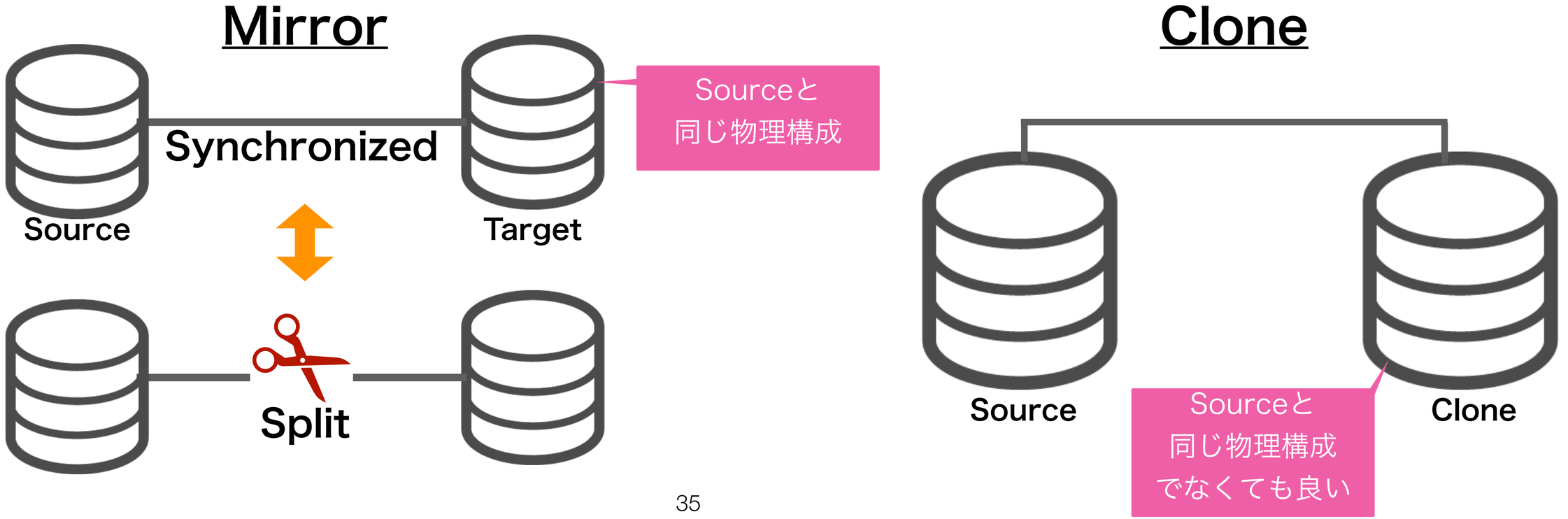
Replicationの種類

- ストレージでは複数種類のReplicationがあり

Type	Local/Remote	Synchronization Type	Memo
Mirror	Local	Sync	
	Local	Async	
	Remote	Sync	
	Remote	Async	
Clone	Local	Sync	
	Local	Async	
	Remote	Sync	
	Remote	Async	
Snapshot	Local	Sync(Full)	世代管理できるClone
	Local	Async(Full)	世代管理できるClone
	Local	Sync(Delta)	
	Local	Async(Delta)	

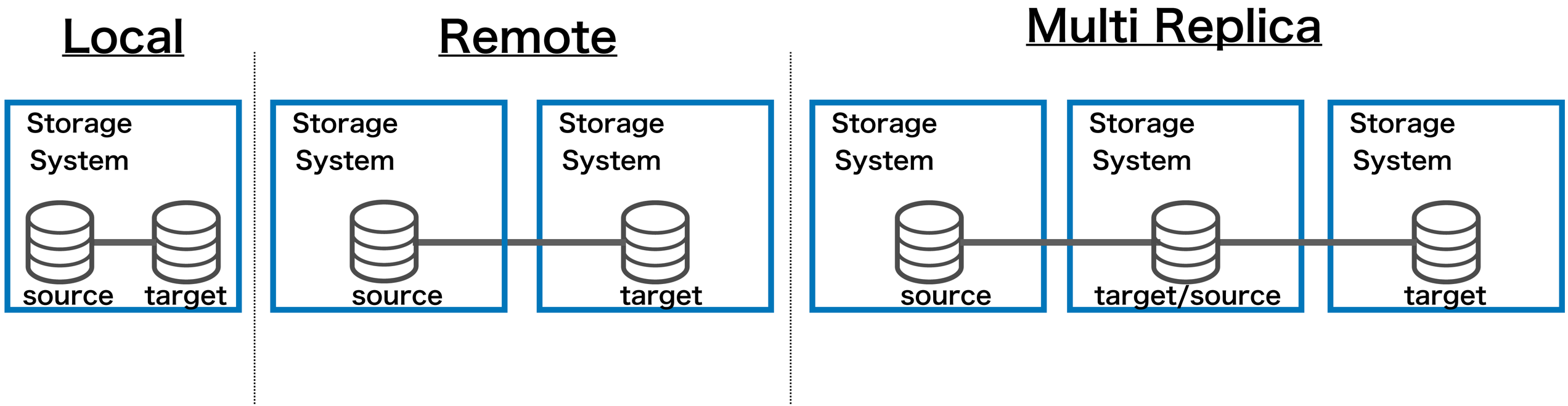
MirrorとClone

- Mirrorは基本はSynchronized(ペア)状態
  - TargeteVolumeにアクセスする場合はSplit
- Cloneは同じ物理構成でなくても可能(ベンダによってはCopy(1回のみのClone))



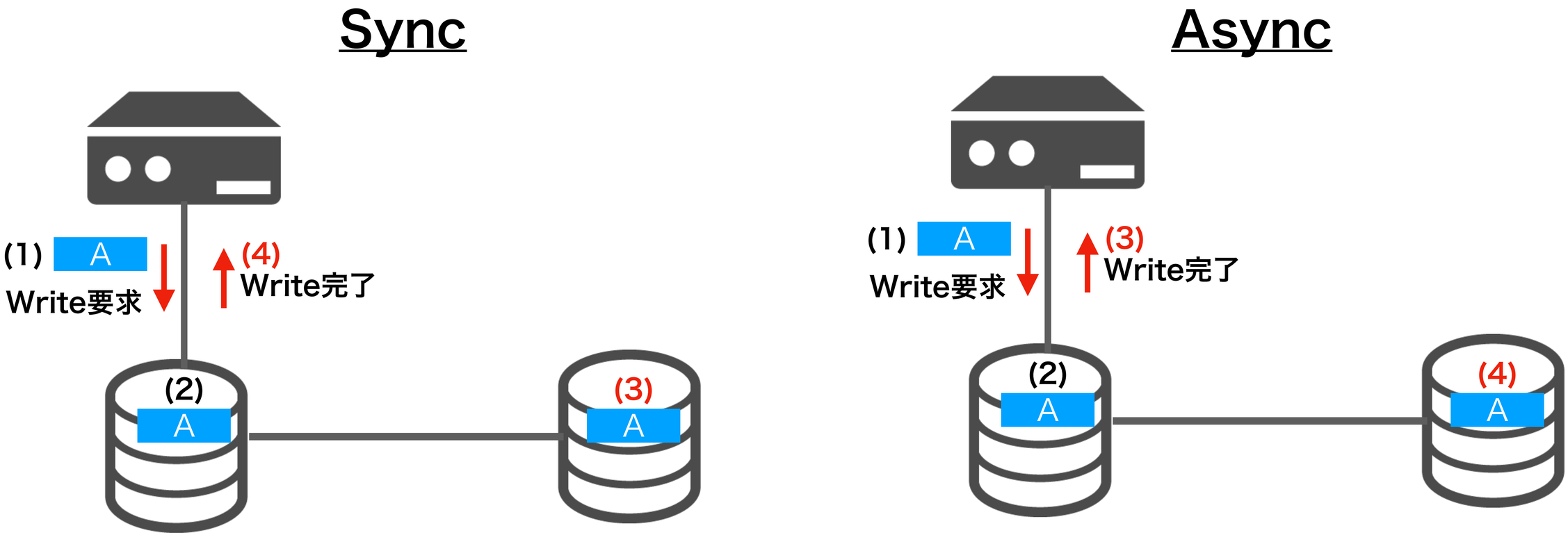
LocalとRemote

- 筐体内(Local)と筐体間(Remote)でMirror/Clone
- 超長距離の場合はMulti Replica 構成



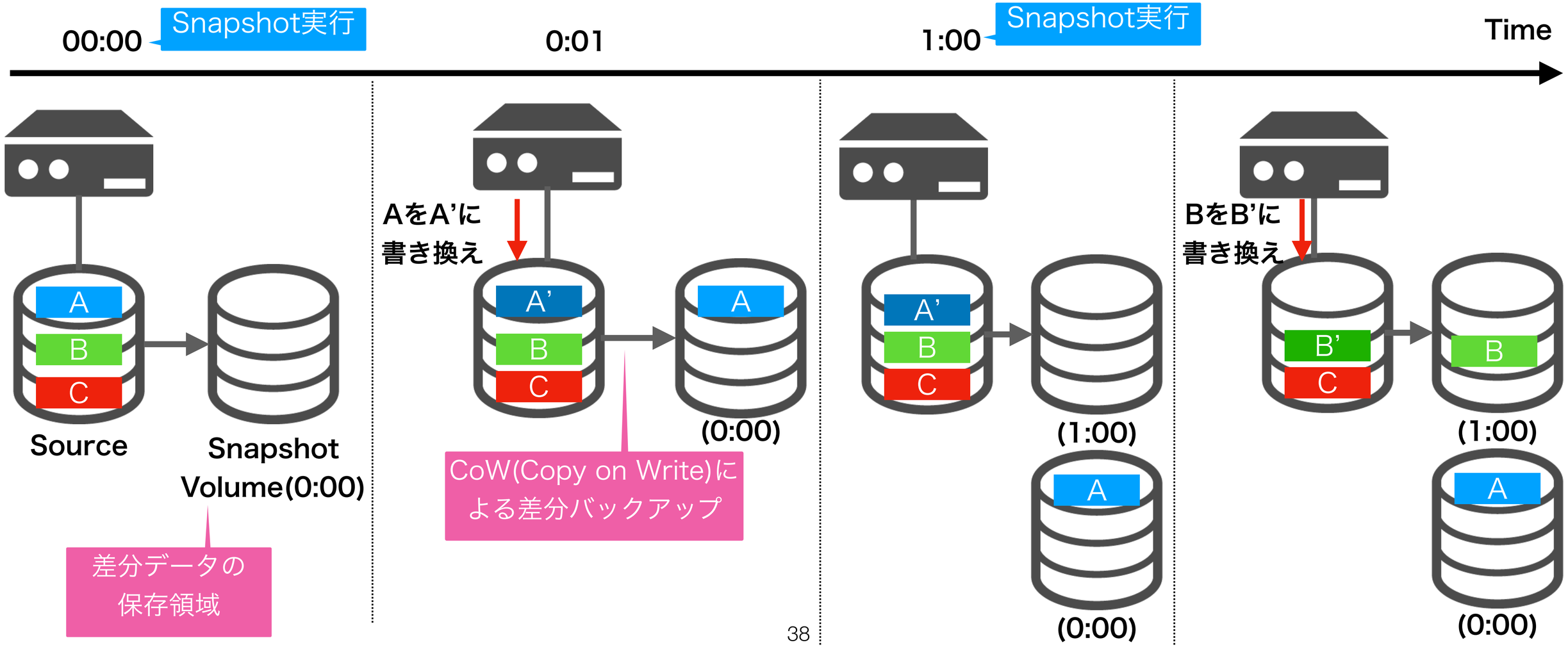
SyncとAsync

- Sync(同期)は近距離間、Async(非同期)は中長距離間で利用
- Asyncはベンダによってデータ転送時に圧縮などを行うものもあり



Snapshot

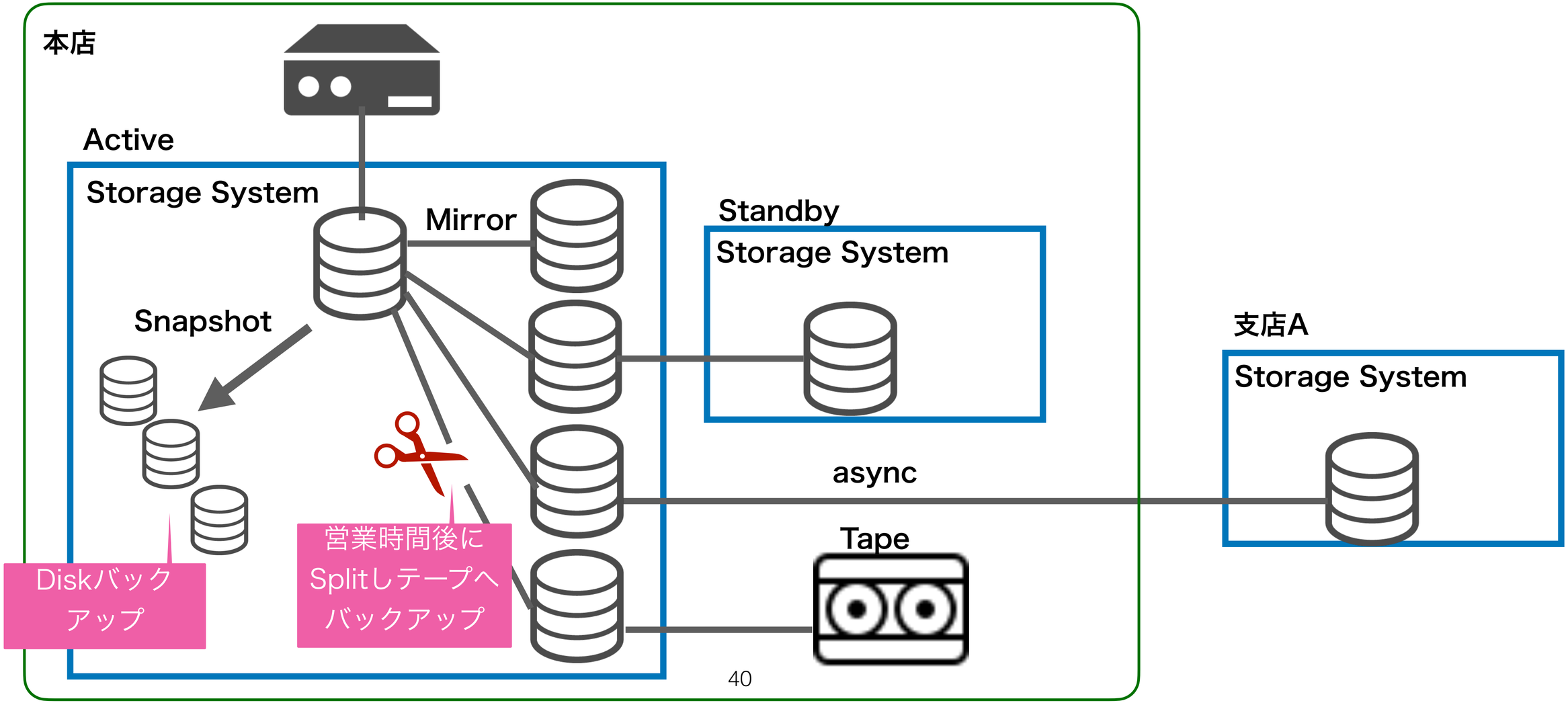
- Snapshotは差分データのみバックアップ



Snapshotの特徴

- 利点
  - 差分バックアップのため消費する物理メディア容量が少ない
  - Thin ProvisioningのLBAマッピングテーブルへの僅かな拡張で実装でき、ストレージ内の管理テーブルの容量削減(Thin Provisioningと相性が良い)
- 欠点
  - Snapshot VolumeからReadする場合、Source Volumeへの性能影響あり
  - CoWによるパフォーマンス劣化がでるベンダもあり
- バックアップ以外のユースケース
  - Snapshot VolumeをWritableにすることでOSのGolden Imageなどでも活用

Replicationの事例



ファイルストレージ

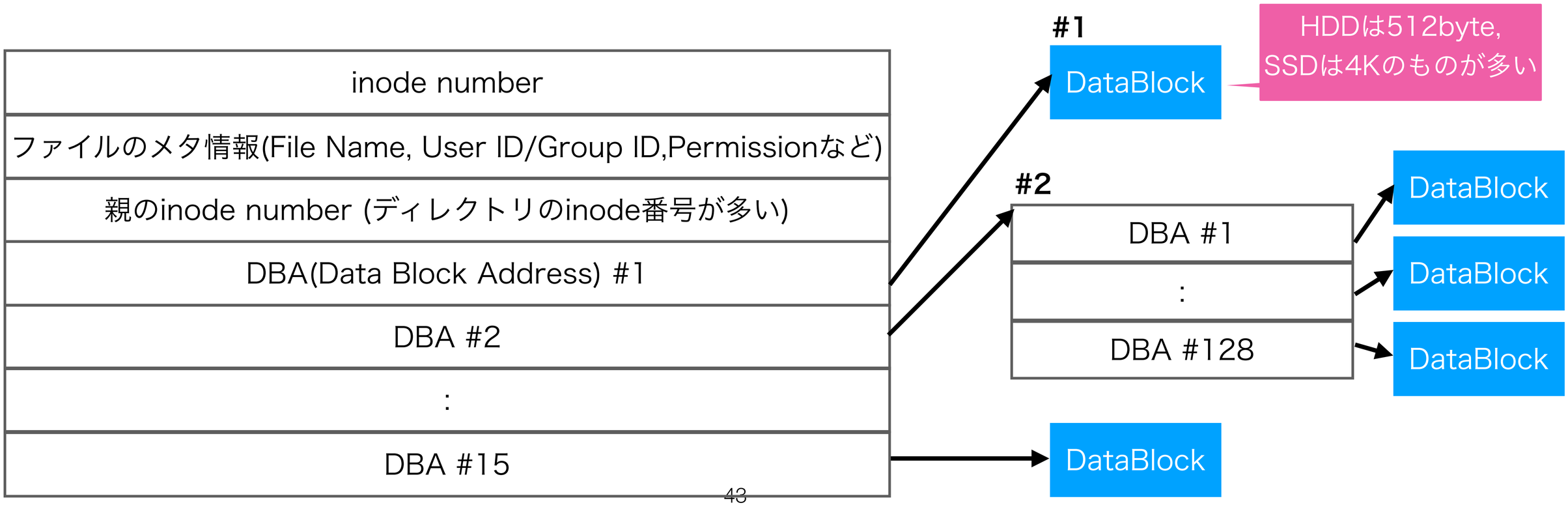
ファイルストレージと歴史

- ファイルストレージ(NAS)はFile Systemを複数サーバで共有するストレージ
- 有名なプロトコルとしてはNFS(Unix系), SMB/CIFS(Windows系)
- NFSはSun Microsystemsが1984年に発表。SunOS上のファイルを共有するプロトコルとして開発。RPC(ONC RPC)はNFSの一部として誕生
  - NFS v3 (TCP or UDP, 2049番Port), NFS v4 (TCP, 2049ポート)
  - 主にNFSd、Network Lock Manager, rquotaから構成される
- SMBは1982年にIBMによって開発された後、マイクロソフトにて独自拡張。CIFSはマイクロソフトにて開発されたSMBベースのオープンなプロトコル
  - 下位プロトコル: SMBはNetBIOS, CIFSはNetBIOS over TCP/IP(445ポート)

NAS: Network Attached Storage, NFS: Network File System, SMB: Server Message Block

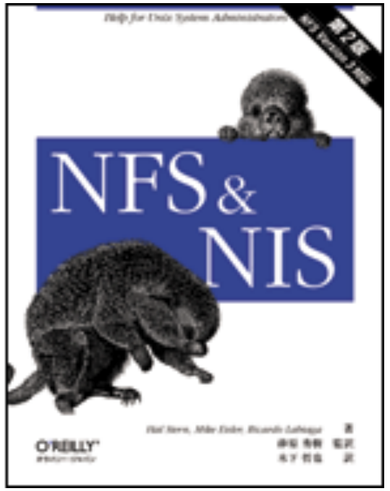
File Systemとは

- OSが提供するファイルを管理するシステム
  - ext4, xfs, zfs, NTFS など
- 基本的なアーキとしてはinodeとData Blockにてファイルを表現



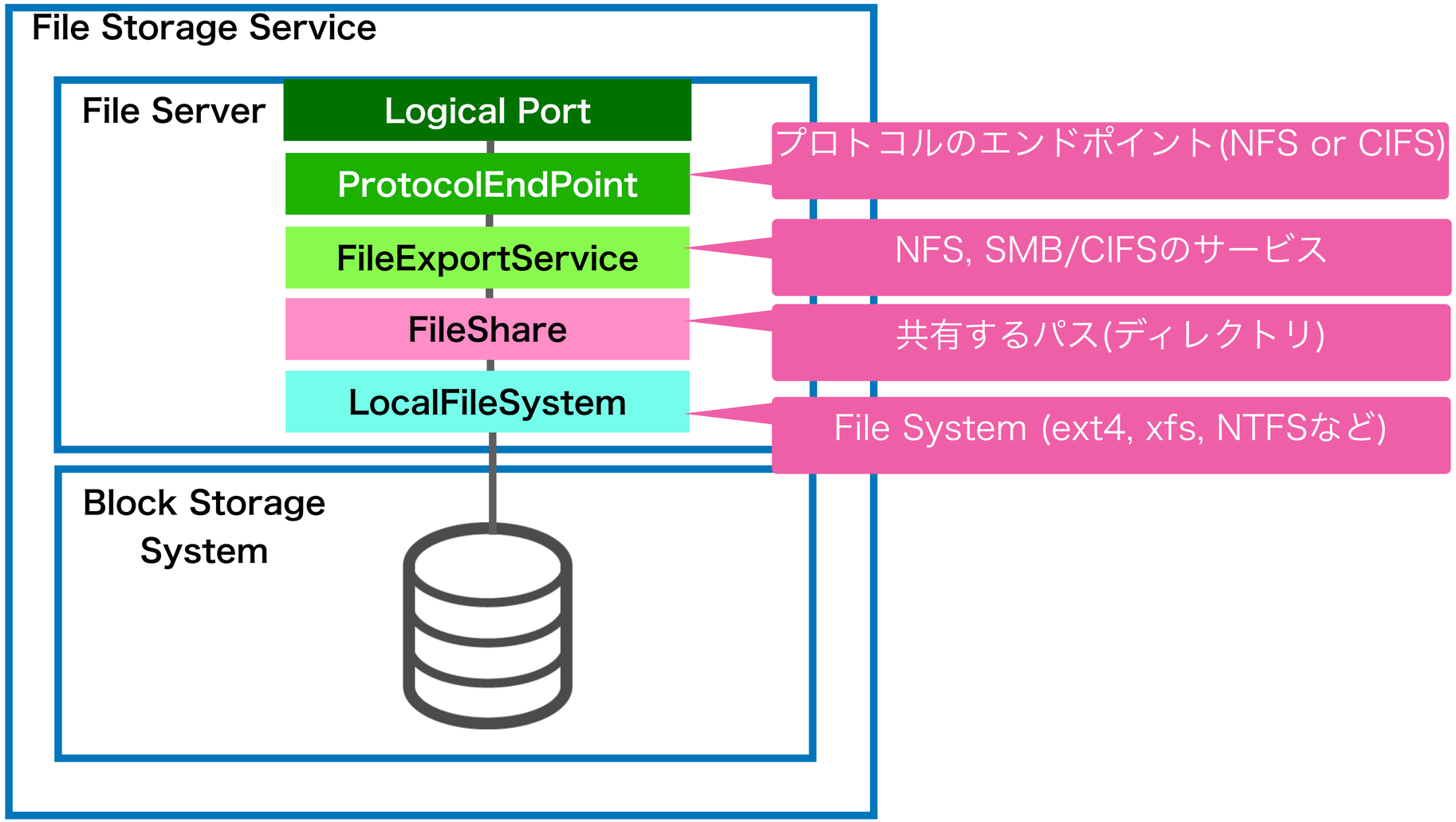
ファイルストレージとは

- File Systemで提供されたファイル(ディレクトリ含む)をNFS, SMB/CIFSなどのプロトコルを通じサーバへ提供
- File Systemをストレージが管理
  - ファイルのUser ID/Group ID, Permissionもストレージの File Systemが管理(アカウント管理とNFSは一緒にシステム設計も多い*)
  - File Systemで実装されているロックやクオータのステータスもNetwork Lock Manager, rquotaにより複数サーバ間で連携



*OREILLY: NFS&NIS

基本構成



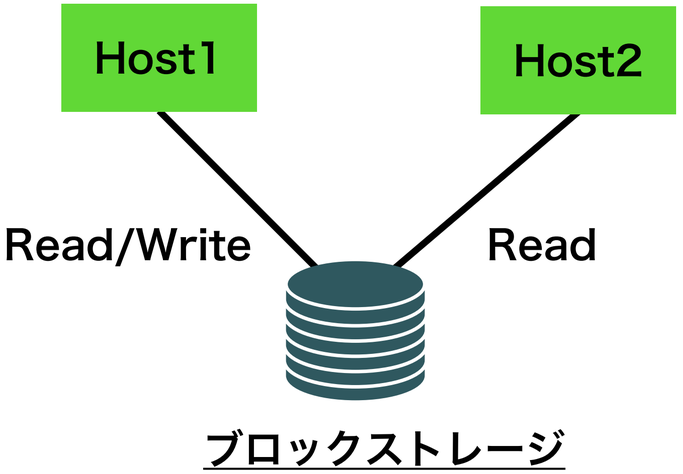
*Block Storage Systemは簡略, File Serverは一般的なLinuxやBSDで実装しているベンダが多い

ファイルストレージの機能

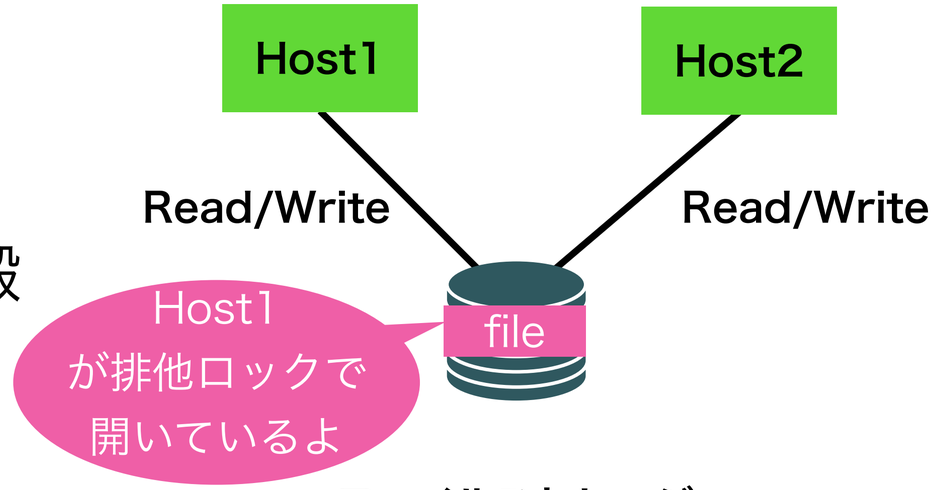
- ストレージとしての機能は、ブロックストレージそのまま
- ファイルストレージ特有の機能
  - アカウント連携 (AD, LDAP連携など)

ロックメカニズムの違い

- ブロックストレージ
  - LBA(論理ブロックアドレッシング)のアクセスのため、ロックメカニズムを持っていない
  - 1台のホストからのみWriteとなるようにマウントする必要あり
- ファイルストレージ
  - ファイル単位でのロックメカニズムあり
  - Open時に取得したロックを他ホストにも伝搬
  - 複数台のホストからRead/Writeが可能



ブロックストレージ



ファイルストレージ

ストレージの雑学

アプライアンスとSDS

- アプライアンスストレージはベンダの専用ハードウェア(独自ASICやCPU, FPGA)+ソフトウェアにて実装
- SDS(Software Defined Storage)はコモディティハードウェア+ソフトウェアにて実装
- 一般的にアプライアンスストレージの方が専用ハードウェアのため高速
- SDSで性能を出すためにはハイスペックなサーバ&台数と高速なネットワークが必要 (各サーバ(Node)間は専用ネットワークが推奨)
  - (例)Ceph(iSCSI)の場合: 6-8 CPUコア、16GBのRAM、2NIC以上、10Gb Ethernetが最低5Node必要(OSDノード: 4, 管理ノード: 1) *

*https://documentation.suse.com/ja-jp/ses/5.5/html/ses-all/storage-bp-hwreq.html

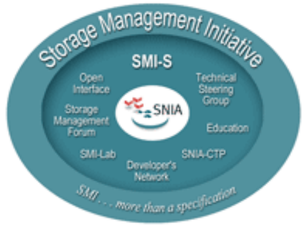
ストレージのプロトコルの進化

- SSDの登場にあわせてデータプロトコルが進化
- 管理モデル/インターフェースは仮想化・クラウドに合わせて進化

	HDD	SSD
メディア		
データプロトコル	FC, iSCSI (SCSI)	NVMe, NVMe-oF
管理モデル/インターフェース	XMLベース (WEBM API)	JSONベース (RESTful API)

新メディア

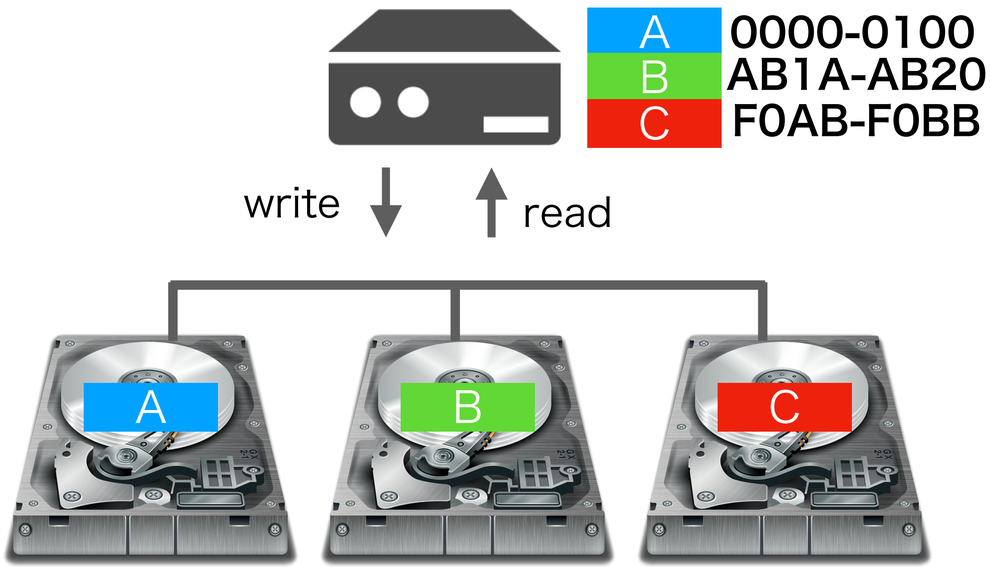
仮想化 クラウド



スループットとIOPS

- ランダムI/Oは1秒あたりのトランザクションが重要なためIOPSが指標
- シーケンシャルI/Oは1秒当たりの処理量が必要なためスループット(MB/sec)が重要

Random I/O



Sequential I/O