iSCSIイニシエータ再入門

日本UNIXユーザ会
山田幸志（だぁやま）
yamada@sdlab.org

InternetWeek2011　仮想化DAY

はじめに

- 海外ではNFS、日本ではiSCSIが流行っている
- 意外にiSCSIの情報が少ない
  - 私はこれで困った。
- 今回はiSCSIを仮想化基盤を利用するという前提でとりあげます。（あまり関係してないが）
  - 運用上必要と思われることを取り扱います
- iscsiのプロトコルや規格は扱いません。
  - 私がそこまで知らないので　m(_ _)m
- **一部粗雑な表現がありますが、ご了承ください**

はじめに

JUS
JAPAN UNIX™ SOCIETY

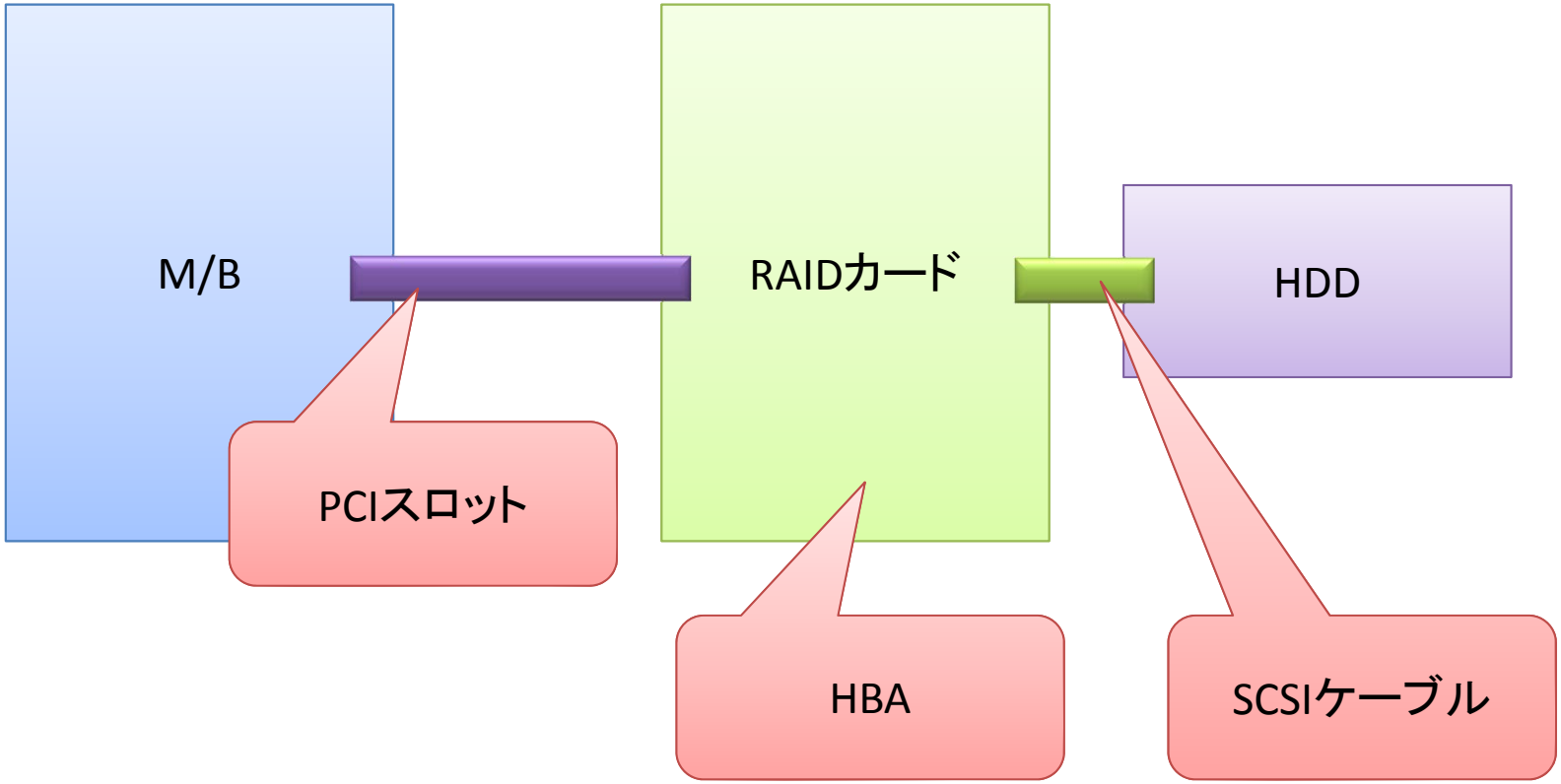
iSCSIとは

- **Internet Small Computer System Interface** (**iSCSI**) は、2003年2月11日にIETFによってRFC (request for comment) として公表された公式な規格への提案 (Proposed standard) であり、SCSIプロトコルをTCP/IPネットワーク上で使用する規格である。
- iSCSIはSCSI-3で規定されるフレームワークではトランスポート層に相当する。トランスポート層には他に並列（パラレル）SCSIやファイバーチャネルがある。（Wikipedia日本語版より）

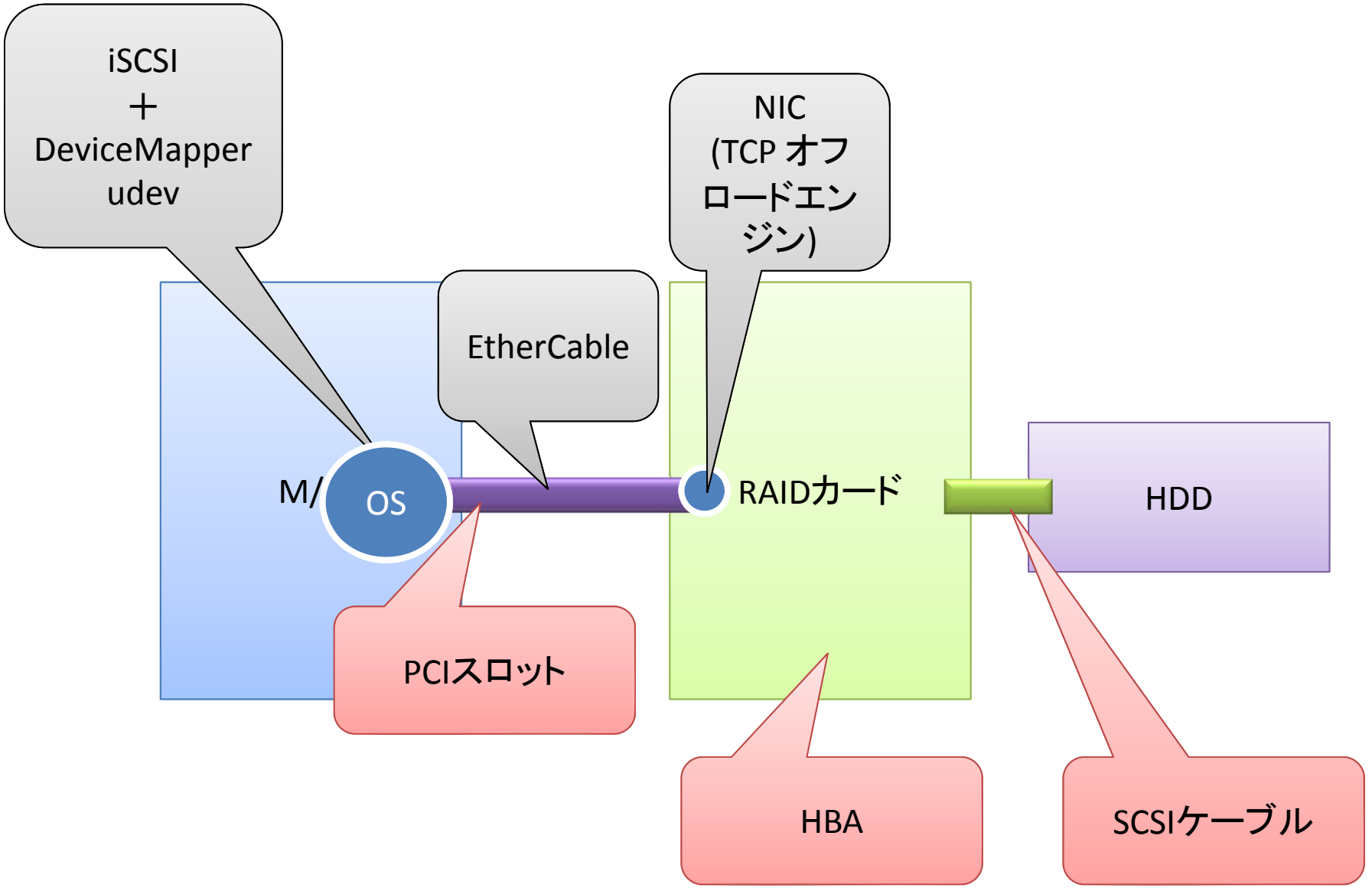
（以下略）

SCSIの確認

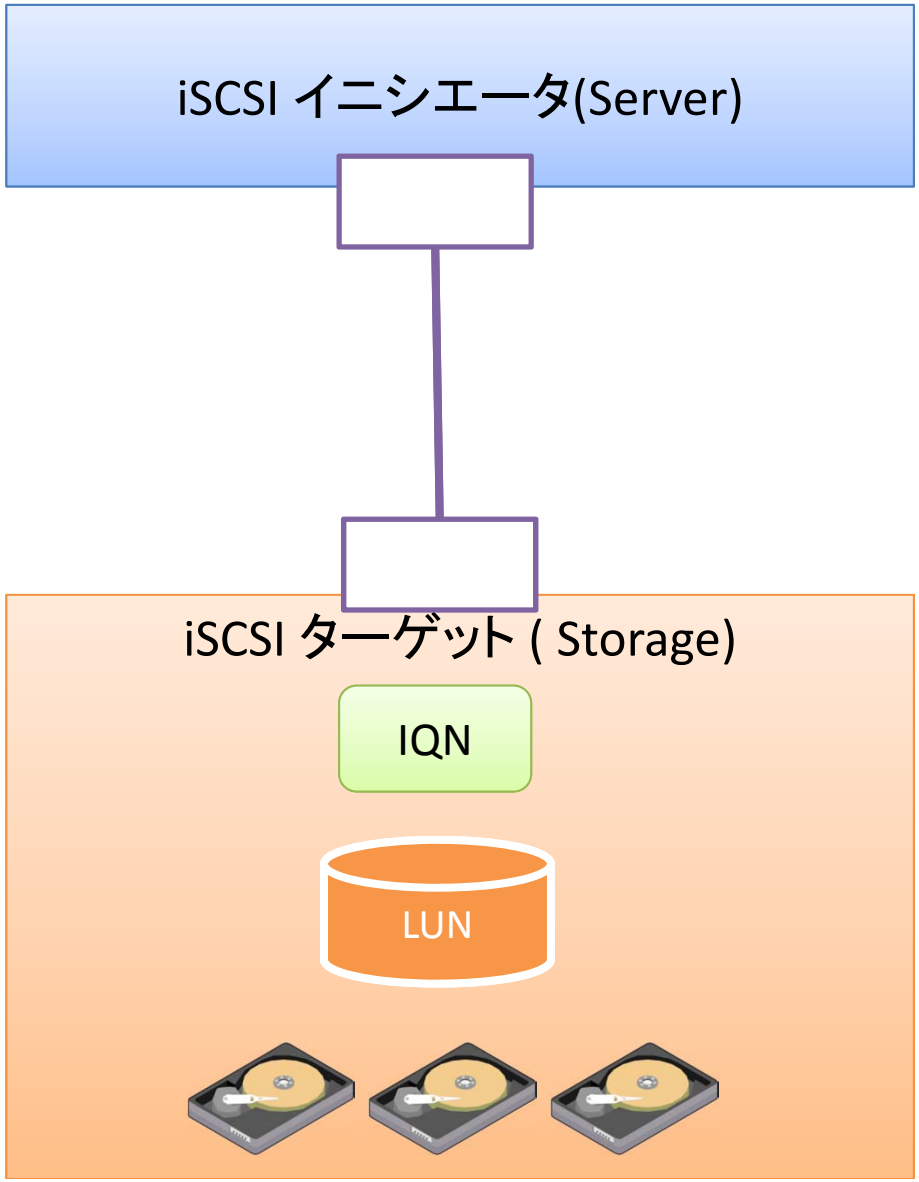
- 基本に戻ろう



SCSIとiSCSI


iSCSI
＋
DeviceMapper
udev
NIC
(TCP オフ
ロードエン
ジン)
EtherCable
M/
OS
RAIDカード
HDD
PCIスロット
HBA
SCSIケーブル

用語の整理



イニシエータ
・利用する方
・いわゆるクライアント
・NFSでいうとmountコマンドを打つ方

ターゲット
・ディスクを持っている方
・いわゆるサーバ

IQN(iSCSI Qualified Name)
・IQNは一意な名前
・書式は違うがMACアドレス的な扱い

IQN

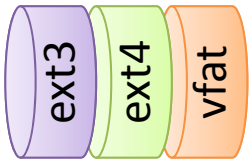
- iSCSI Enterprise Target Daemonの/etc/ietd.conf例で説明

```
Target iqn.1983-06.jp.or.jus:storage.hdb (IQN設定)
IncomingUser
OutgoingUser     (ユーザ認証なし)
Alias iSCSI_Test     (このDISKに名前をつける)
Lun 0 Path=/dev/hdb,Type=fileio
```

- IQNの命名規則
  - ちゃんとあります。従ってください。
    - でも、テストやClosedな環境では適当につけても問題ないと考えています。
  - RFC3721

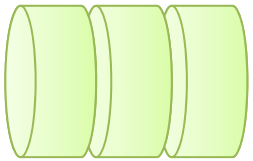
iSCSIの位置づけ

OS



ファイルシステム

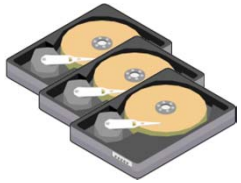
NAS（NFS)



パーティション

LUN(Logical Unit Number)
OSからHDDとして認識する塊

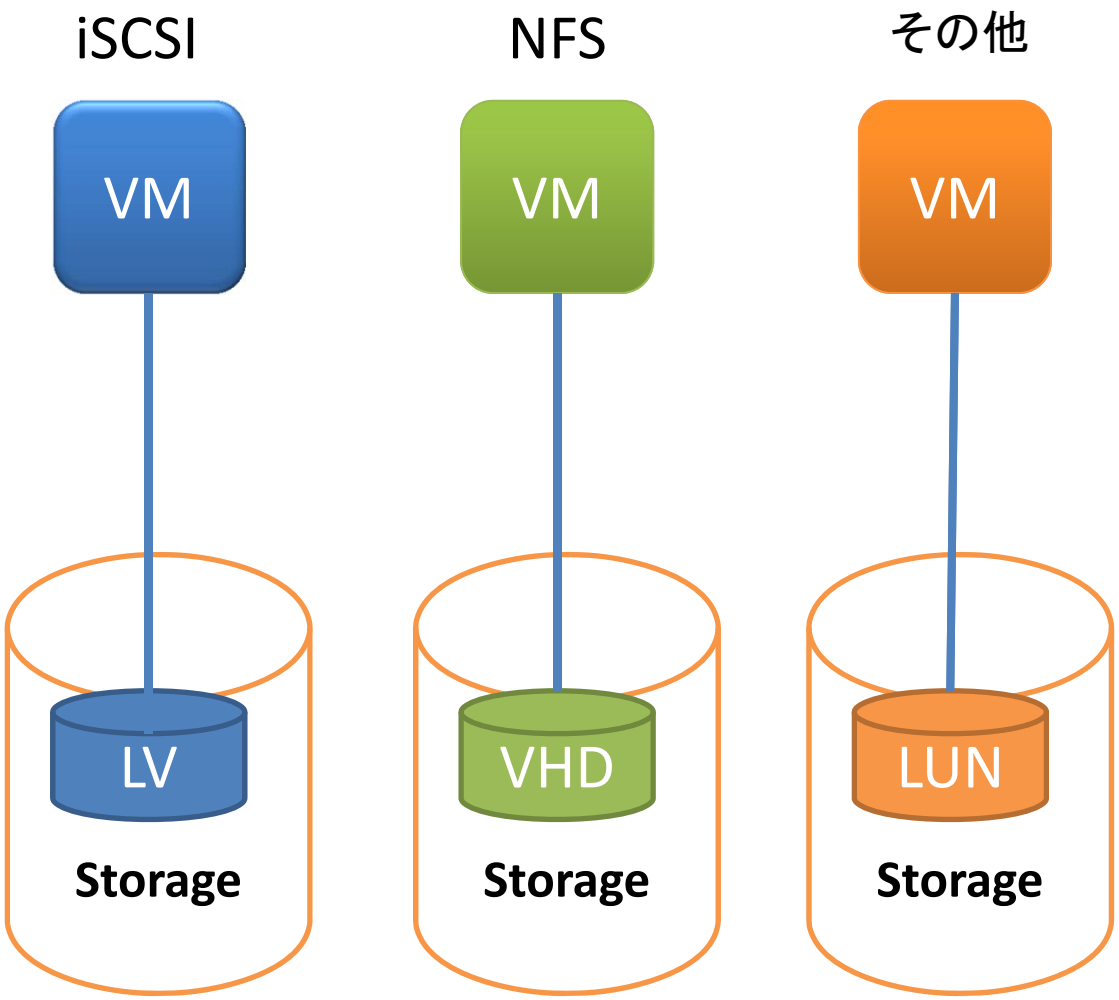
iSCSI
Local HDD
USB



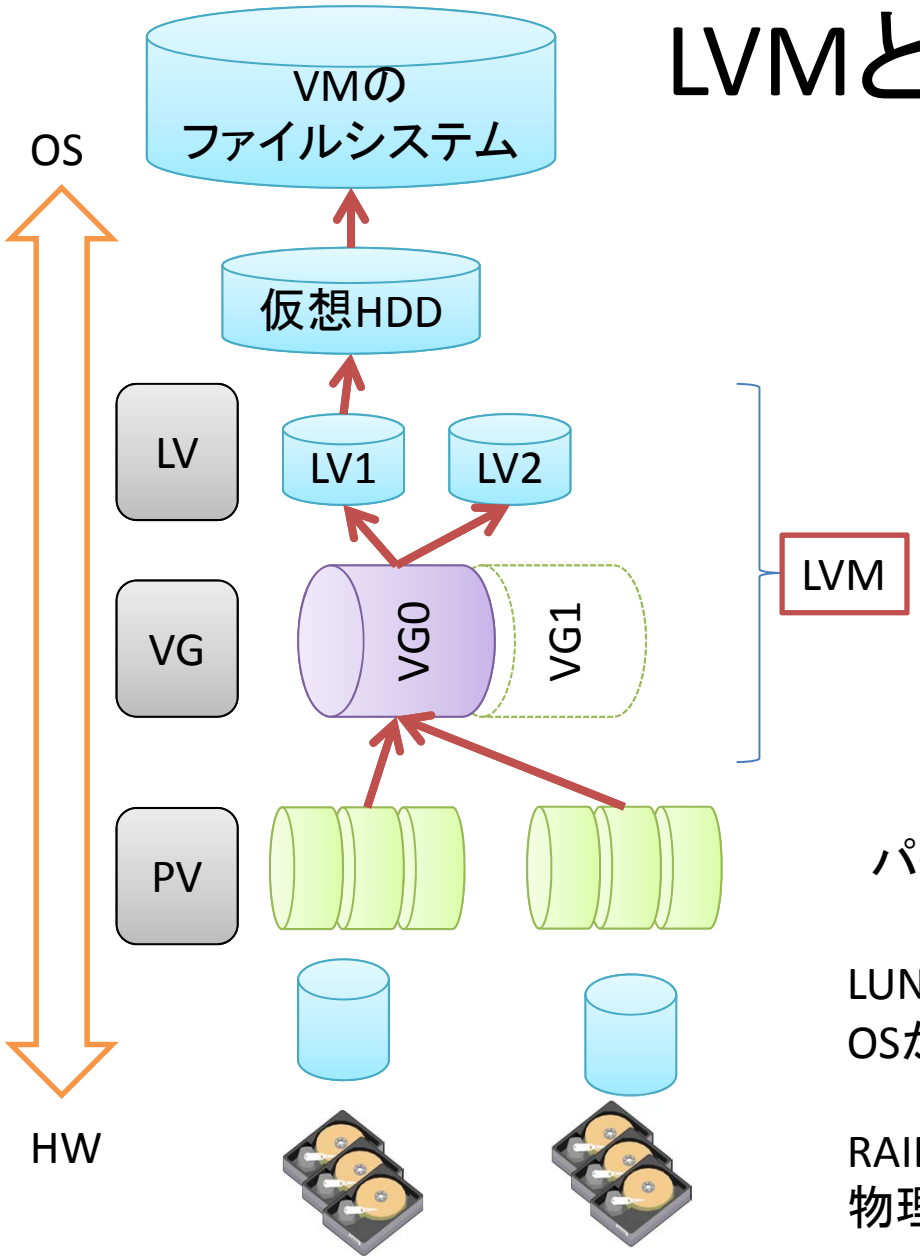
RAIDグループ
物理HDDの集合体を論理的なHDDとしたもの

HW

仮想化基盤での使われ方（例）



仮想化基盤での使われ方
LVMとiSCSI



/dev/VolGroup00/LogVol00
/dev/VolGroup00/LogVol01

/dev/VolGroup00
/dev/VolGroup01

パーティション

/dev/sda1
/dev/sda2

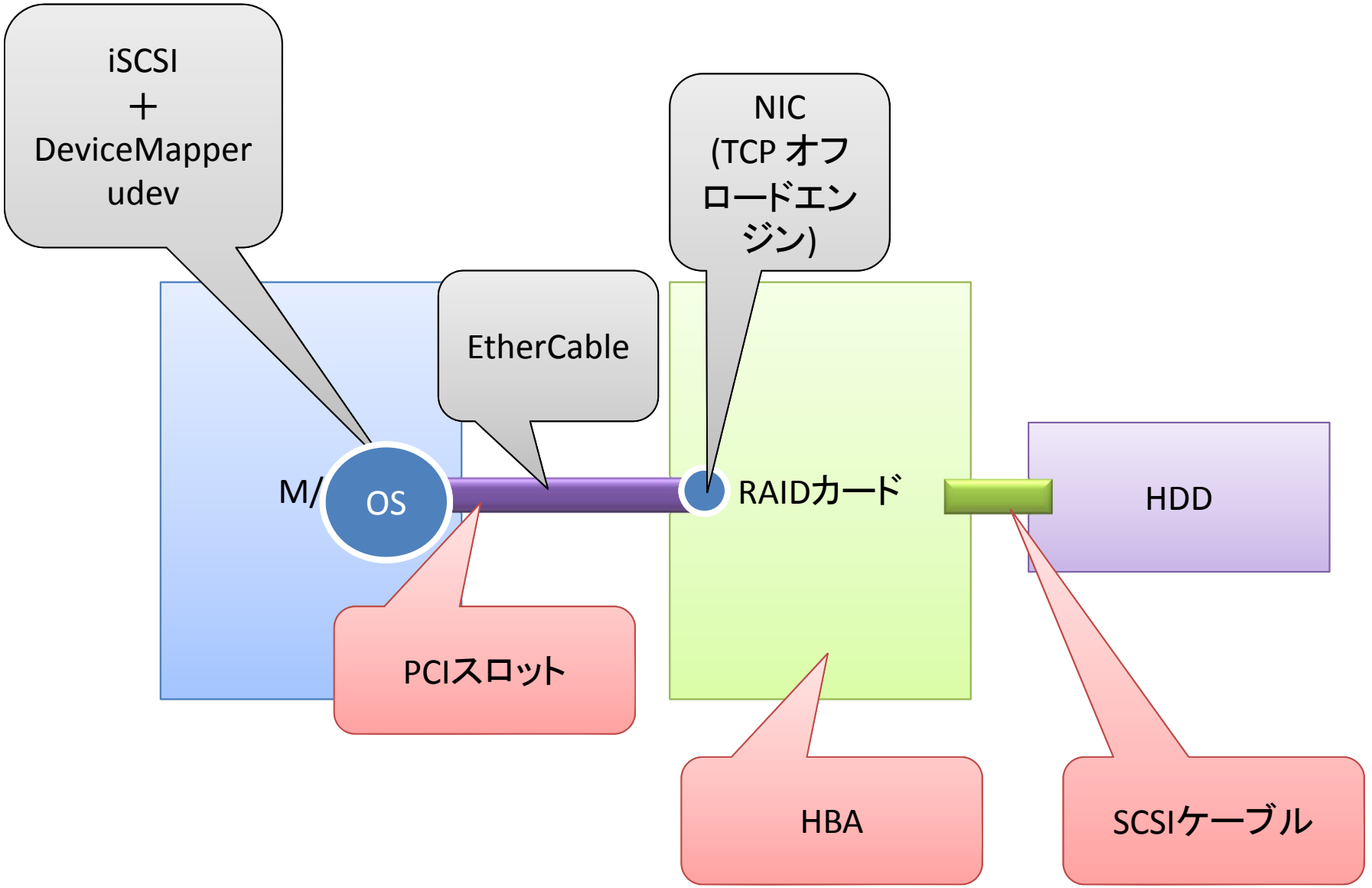
LUN(Logical Unit Number)
OSからHDDとして認識する塊

/dev/sda
/dev/sdb

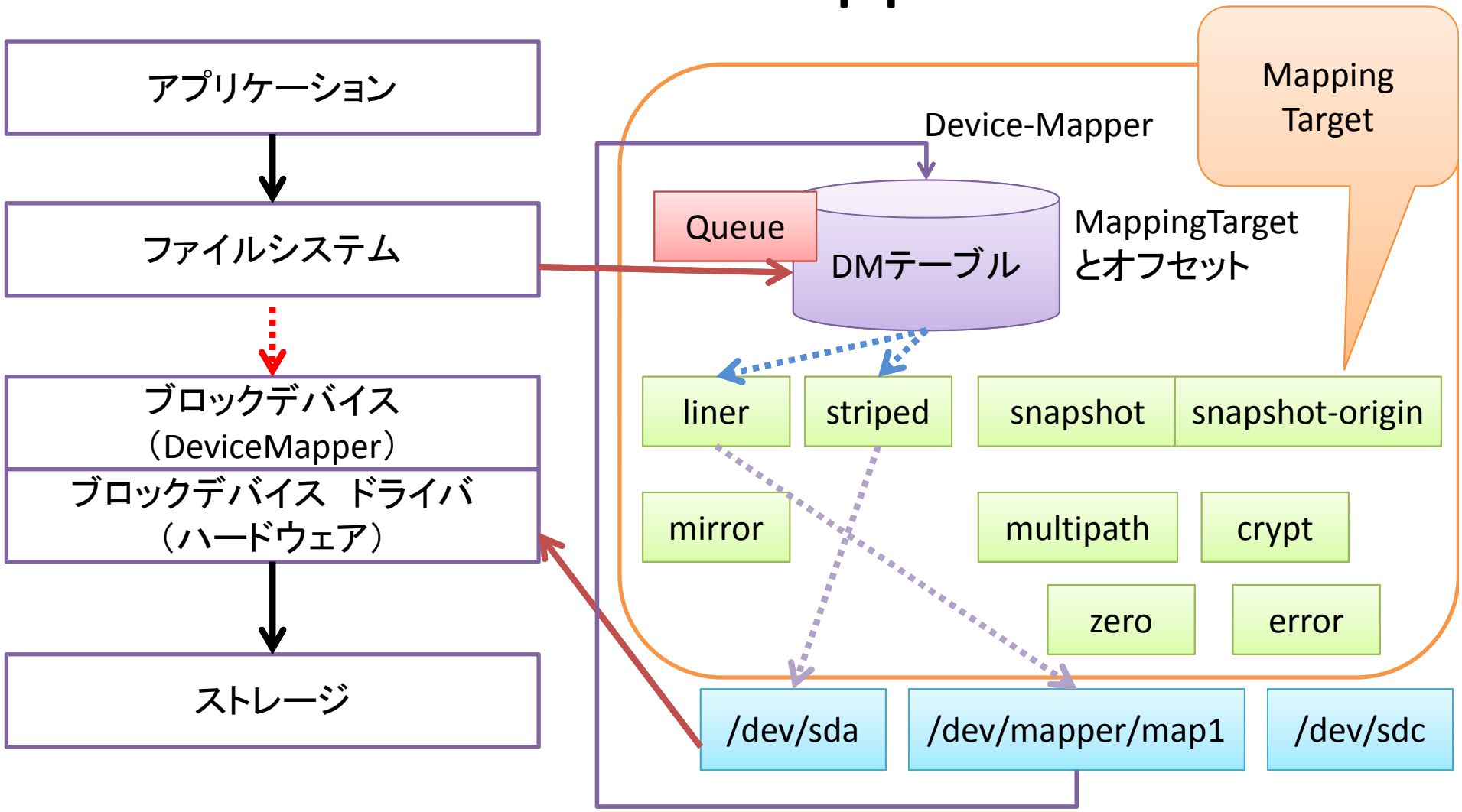
RAIDグループ
物理HDDの集合体を論理的なHDDとしたもの

DEVICE-MAPPER

SCSIとiSCSI


iSCSI
＋
DeviceMapper
udev
NIC
(TCP オフ
ロードエン
ジン)
EtherCable
M/
OS
RAIDカード
HDD
PCIスロット
HBA
SCSIケーブル

Device-Mapper



udev

- USB-HDDやUSBメモリみたいなものをつなぐ時便利。
- udevは動的にデバイスを管理する仕組み
- udevにおいては各種デバイスの取り扱い方を、/etc/udev/rules.d/配下のルールファイルに記述

00-backdev.rules 55-xs-mpath-scsidev.rules 60-raw.rules xen-backend.rules
05-udev-early.rules 56-emulex-bsg.rules 90-dm.rules xen-frontend.rules
40-multipath.rules 56-emulex-bsg.rules.bak 90-hal.rules z10-xen-vcpu-hotplug.rules
50-udev.rules 58-xapi.rules 95-pam-console.rules
51-hotplug.rules 60-net.rules blktap.rules

ホットプラグ

- ホットプラグデバイス接続時などに利用
  - 以前は/sbin/hotplug
  - 最近はudev
- 流れ
  - デバイス接続
  - udevdに通知
  - /etc/udevdのスクリプト実効
  - haldに通知

/dev/mapperの状態

Centos

```
[root@centos53x64 ~]# ls -al /dev/mapper/
total 0
drwxr-xr-x  2 root root     100 Nov 25 12:20 .
drwxr-xr-x 11 root root    2900 Nov 25 12:42 ..
crw-------  1 root root  10, 62 Nov 25 12:20 control
brw-rw----  1 root disk 253,  0 Nov 25 12:20 VolGroup00-LogVol00
brw-rw----  1 root disk 253,  1 Nov 25 12:20 VolGroup00-LogVol01
```

XenServer + Multipath

```
[root@test01 udev]# ls -al /dev/mapper
total 0
drwxr-xr-x  2 root root     120 Nov 25 12:06 .
drwxr-xr-x 18 root root   17060 Nov 25 12:06 ..
brw-rw----  1 root disk 252,  0 Nov 25 12:06 3600000e00d0000000030538000000000
crw-------  1 root root  10, 56 Nov 17 15:00 control
brw-rw----  1 root disk 252,  2 Nov 25 12:12
VG_XenStorage--5735ad14--3871--5243--5d98--f0394db7a007-VHD--487a7522--dd37--
brw-rw----  1 root disk 252,  1 Nov 25 12:12
VG_XenStorage--5735ad14--3871--5243--5d98--f0394db7a007-VHD--c36d4e03--25d5--
```

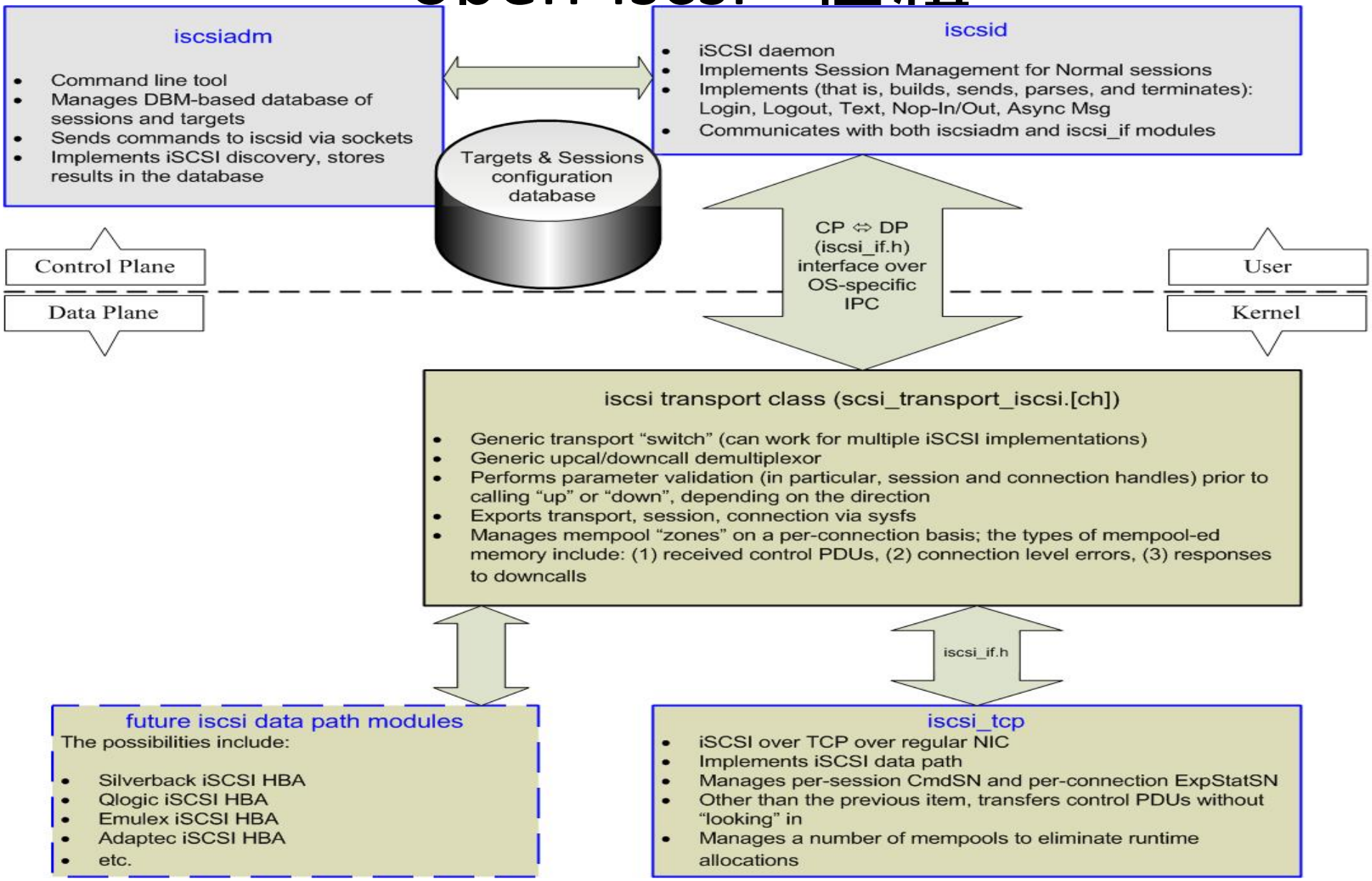
OPEN-ISCSI

open-iscsi

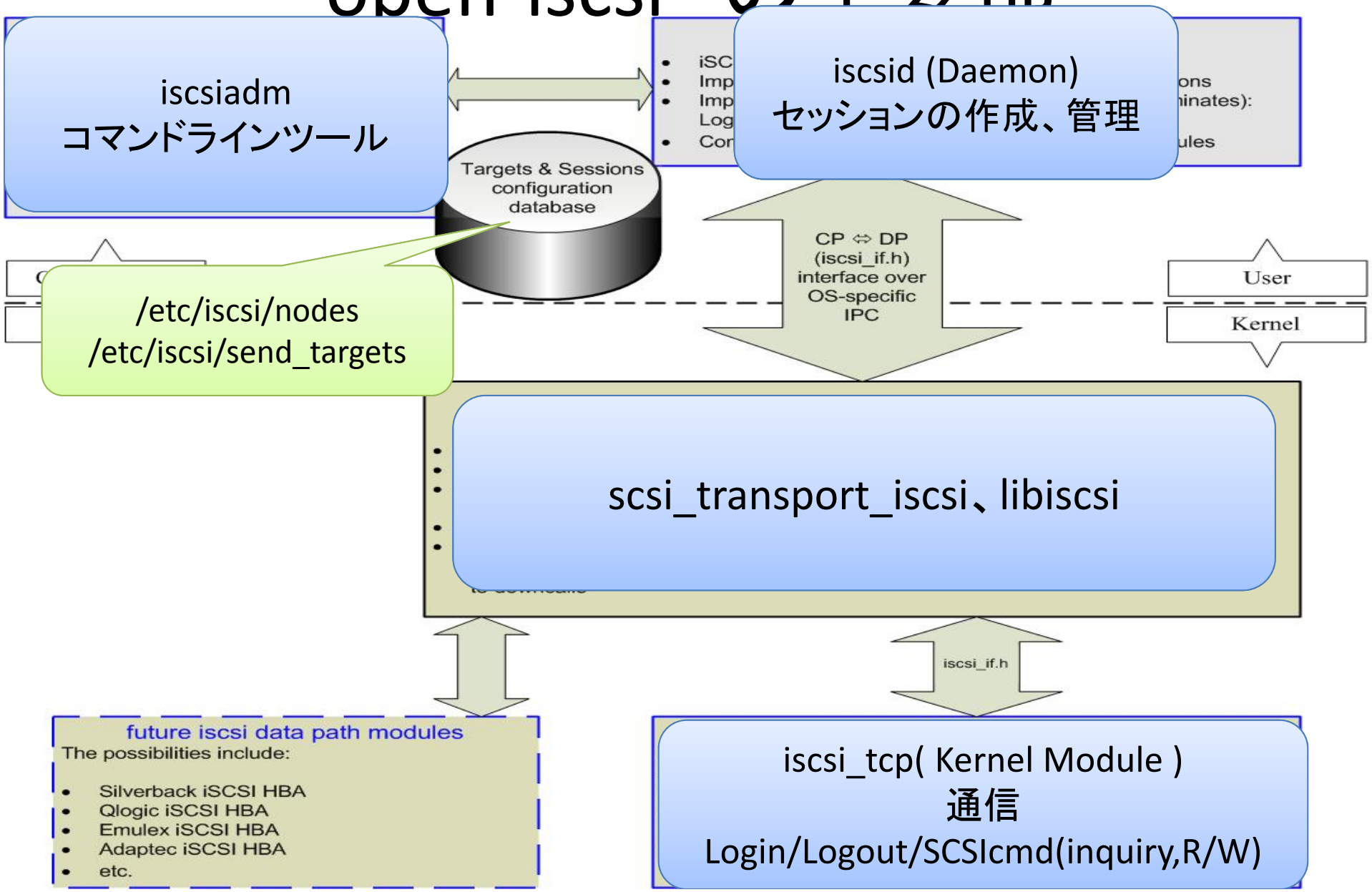
- OpenSourceなiSCSIイニシエータ
- 主な構成
  - CLIツールのiscsiadm
  - 管理Daemon

出典：www.open-iscsi.org

open-iscsi 仕組



open-iscsi の主要部



open-iscsi コンフィグ例

- **iscsid.conf**

  node.startup = manual
  node.session.timeo.replacement_timeout = 15
  node.conn[0].timeo.login_timeout = 15
  node.conn[0].timeo.logout_timeout = 15
  node.conn[0].timeo.noop_out_interval = 10
  node.conn[0].timeo.noop_out_timeout = 15
  node.session.initial_login_retry_max = 4
  node.session.cmds_max = 128
  node.session.queue_depth = 32
  （以下略）

- **initiatorname.scsi**

  InitiatorName=iqn.1983-06.jp.or.jus:cafe1234
  InitiatorAlias=test01

open-iscsiを使ってみる

- open-iscsiに関してはREADMEが一番情報がある。

- 大まかな流れ
  - 接続先（ターゲット）を見つける
  - nodesに登録される
  - 登録されたnodeにログインする
  - ブロックデバイスとして使える

open-iscsi コマンド

- セッション検索
  - # iscsiadm -m discovery --type sendtargets --portal 192.168.13.1
- ログイン
  - # iscsiadm -m node -T <IQN> -p 192.168.13.1 -l
- ログアウト
  - # iscsiadm -m node -T <IQN> -p 192.168.13.1 --logout
- ログイン（検索し認識しているセッション全てにログイン）
  - # iscsiadm -m node -L all
- **セッション情報**
  - 現在張られているセッションの詳細情報
  - # cat /etc/iscsi/nodes/iqn.2000-09.s50:storage-kudo:maz:tomop/192.168.13.1¥,3260¥,1/default
- **セッションとDISKの紐づきを確認する**
  - # iscsiadm -m session -P 3

iscsiadm

複数セッションの例

```
[root@test01 ~]# iscsiadm -m node
192.168.1.1:3260,1 iqn.1983-06.jp.or.jus:storage-system:00030538cm00
192.168.1.2:3260,3 iqn.1983-06.jp.or.jus:storage-system:00030538cm10
192.168.2.1:3260,2 iqn.1983-06.jp.or.jus:storage-system:00030538cm01
192.168.2.2:3260,4 iqn.1983-06.jp.or.jus:storage-system:00030538cm11

[root@test01 ~]# iscsiadm -m node -T iqn.1983-06.jp.or.jus:storage-system:0003053
-p 192.168.1.2 --login

Logging in to [iface: default, target: iqn.1983-06.jp.or.jus:storage-system:00030538cm10
, portal: 192.168.1.2,3260]
Login to [iface: default, target: iqn.1983-06.jp.or.jus:storage-system:00030538cm10,
 portal: 192.168.1.2,3260]: successful

[root@test01 ~]# iscsiadm -m node -L all
```

確認

```
[root@test01 ~]# iscsiadm -m session
tcp: [1] 192.168.1.1:3260,1 iqn.1983-06.jp.or.jus:storage-system:00030538cm00
tcp: [2] 192.168.1.2:3260,3 iqn.1983-06.jp.or.jus:storage-system:00030538cm10
tcp: [3] 192.168.2.1:3260,2 iqn.1983-06.jp.or.jus:storage-system:00030538cm01
tcp: [4] 192.168.2.2:3260,4 iqn.1983-06.jp.or.jus:storage-system:00030538cm11
```

iscsiadm –m session –P 3

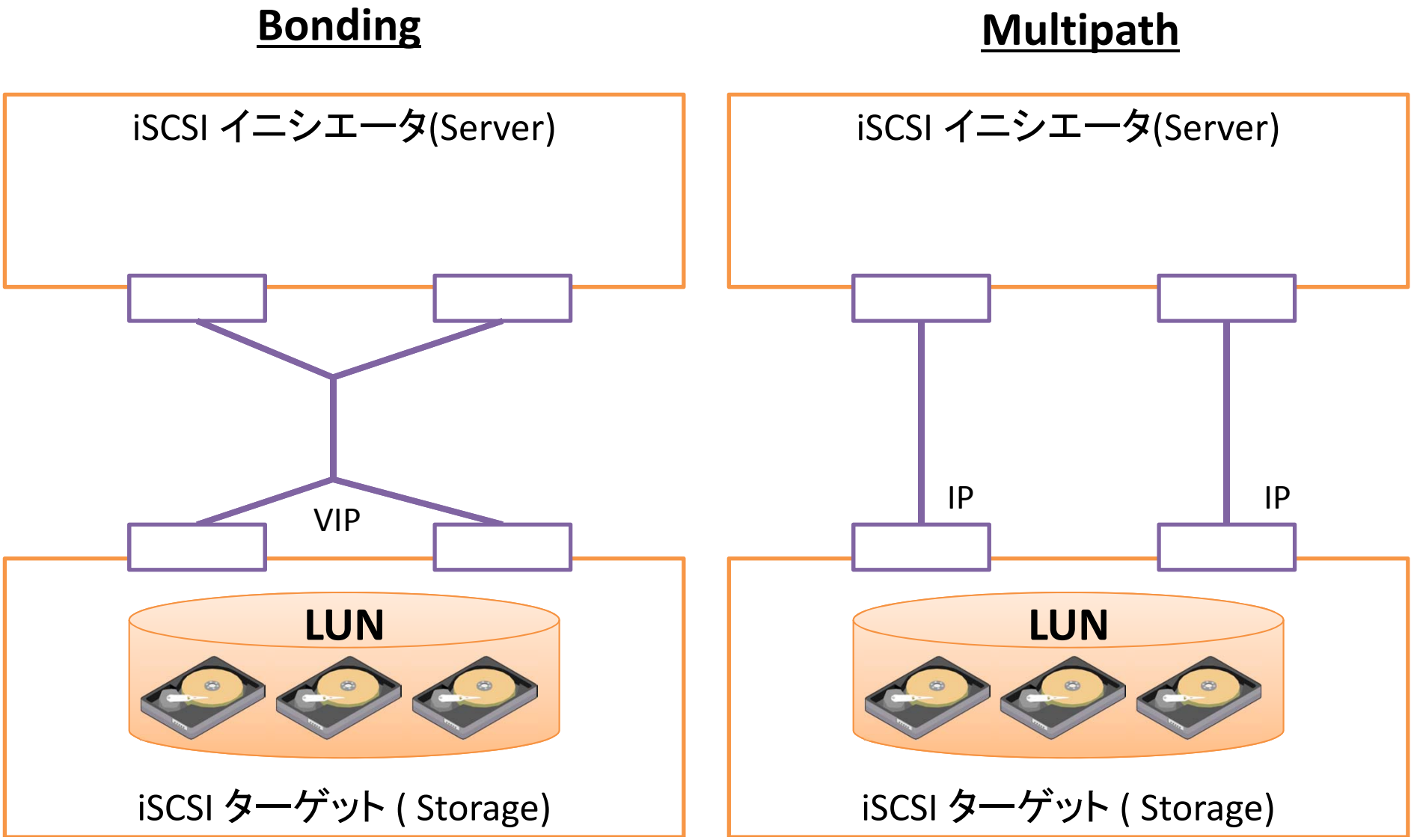
```
[root@test01 ~]# iscsiadm -m session -P 3
iSCSI Transport Class version 2.0-870
iscsiadm version 2.0-870
Target: iqn.1983-06.jp.or.jus:storage-system:00030538cm00
        Current Portal: 192.168.13.1:3260,1
        Persistent Portal: 192.168.13.1:3260,1
                **********
                Interface:
                **********
                Iface Name: default
<省略>
                ************************
                Attached SCSI devices:
                ************************
                Host Number: 5  State: running
                scsi5 Channel 00 Id 0 Lun: 0
                        Attached scsi disk sdb          State: running
                scsi5 Channel 00 Id 0 Lun: 1
                        Attached scsi disk sdc          State: running
```

ログアウトの注意点

- iscsiadm でログアウトすると、当然セッションが切れます。
  - セッションが0本＝HDDがない
  - 仮想化基盤で使ってたら大変な事態に。
  - 障害で切れてもいっしょです。

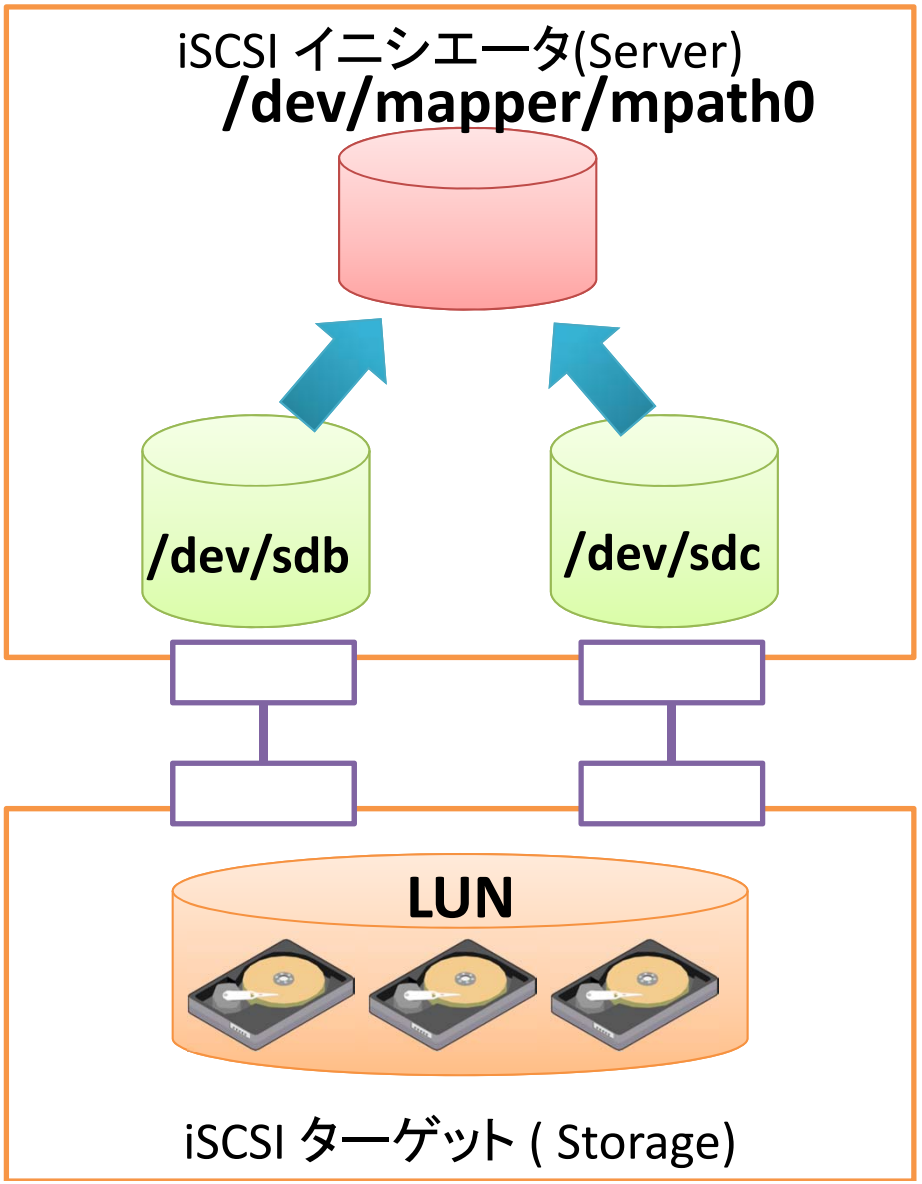
冗長化

iSCSIの冗長化


Bonding
iSCSI イニシエータ(Server)
VIP
LUN
iSCSI ターゲット (Storage)
Multipath
iSCSI イニシエータ(Server)
IP
IP
LUN
iSCSI ターゲット (Storage)

MULTIPATH-TOOLS

Multipathとは



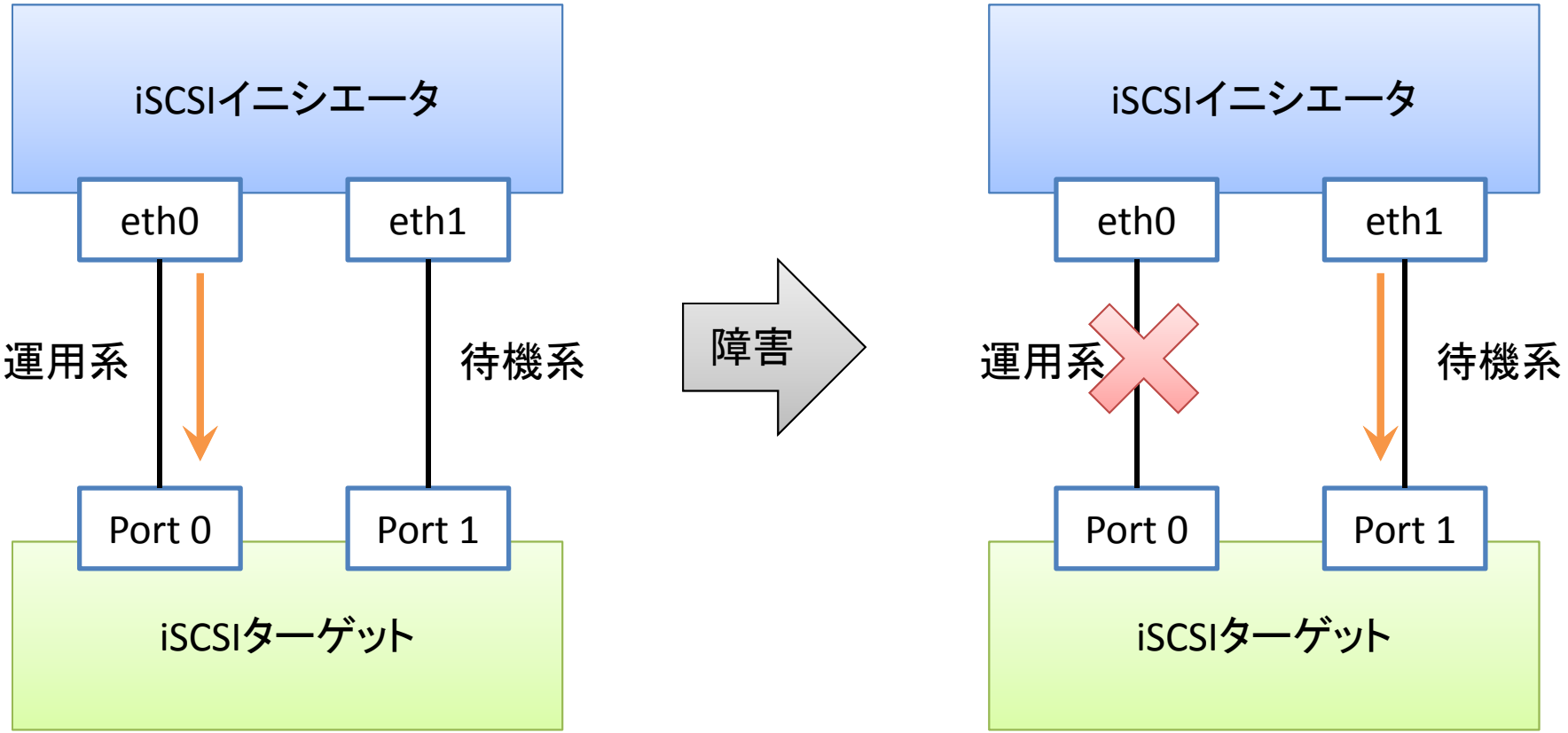
- マルチパスI/O
- device-mapperの機能の1つ
  - ディスク実体に対して、複数のパスを張れる場合、これらを一つの DM デバイスとして扱える機能
  - FailOverやI/Oの振り分けも可能

Multipathの種類

- FailOver構成
  - Active / Standbyの構成
  - 1pathがActiveで、その他がStandby
  - Activeが切れた場合に、切替に時間がかかる（iSCSIイニシエータの問題）
- MultiBus構成
  - 複数のPathを利用する構成
  - 全部のPathがActive。
  - 複数Path中、数Path切れても継続して利用可能
    - 切れたPathの処理に時間がかかる（iscsiイニシエータの問題）
- その他
  - group_by_serial = 検出されたシリアル番号毎に1つの優先グループ
  - group_by_prio = パス優先値毎に1つの優先グループ
  - group_by_node_name = ターゲットノード名毎に1つの優先グループ

MultipathのFailover

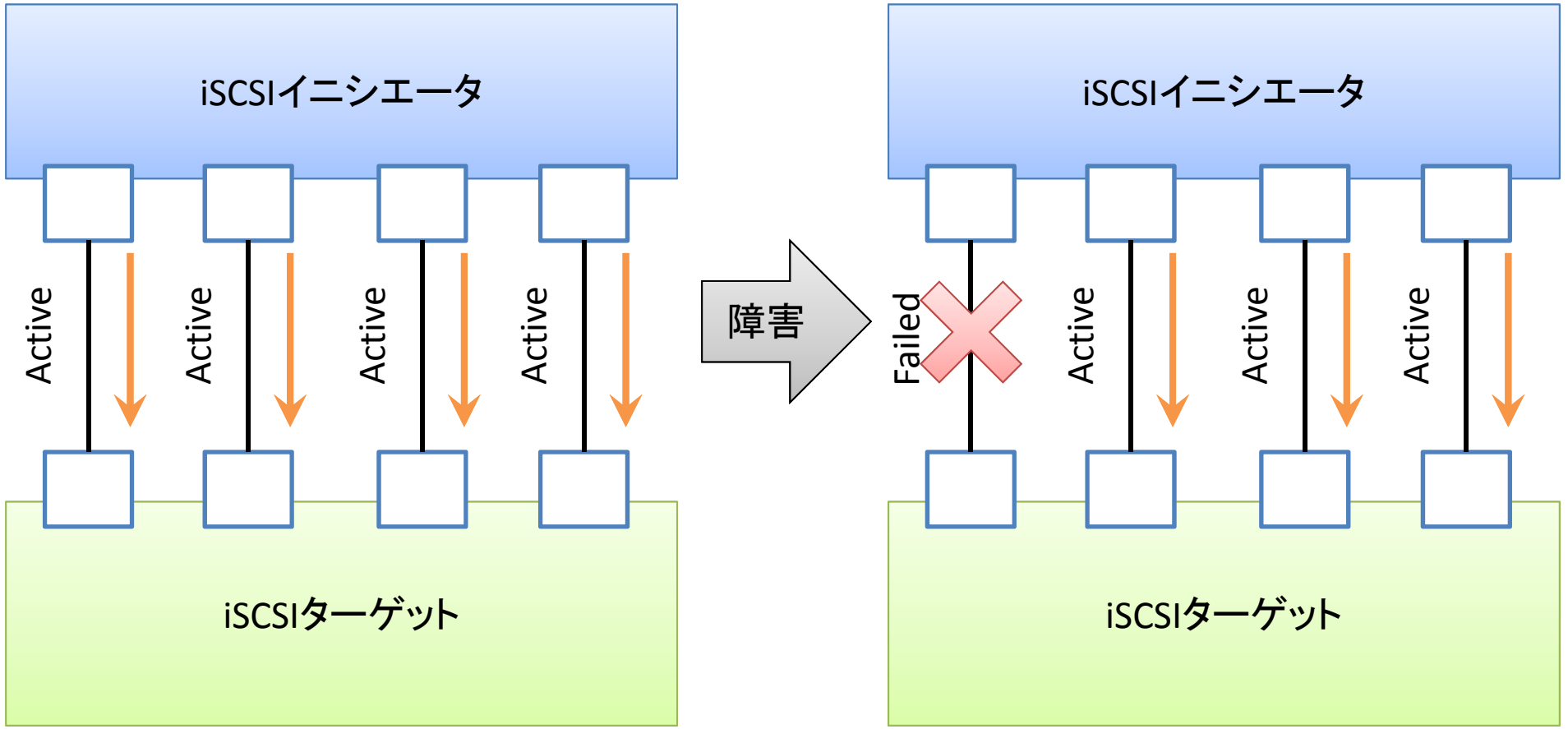
運用系で障害が発生すると、待機系に切り替わる



MultipathのMultiBus

全てのPathを利用する。負荷分散型。

障害時
他の3本を利用して通信される

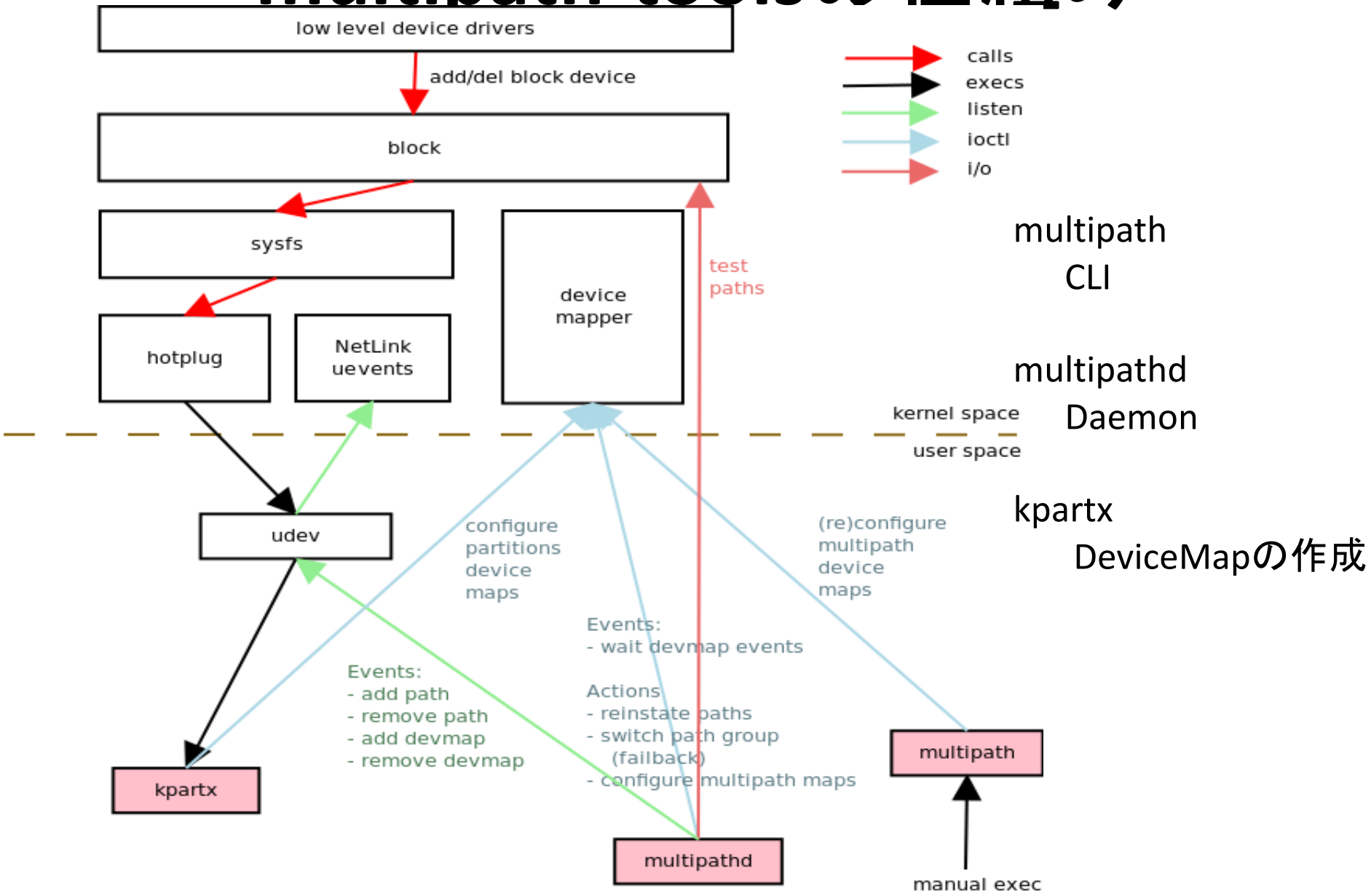


multipath-tools

- http://christophe.varoqui.free.fr/
- GPLライセンス
- DeviceMapperを利用してMultipathを使えるようにするとツール群

出典：http://christophe.varoqui.free.fr/

multipath-toolsの仕組み



multipath
CLI

multipathd
Daemon

kpartx
DeviceMapの作成

multipath-tools コンフィグ

- 省略m(_ _)m
- 注釈つきコンフィグ
  - multipath.conf.annotate
- 詳しいコンフィグの説明
  - http://docs.redhat.com/docs/ja-JP/Red_Hat_Enterprise_Linux/6/html/DM_Multipath/mpio_configfile.html
  - ※Kernelが新しいので動作の差異に注意！
- ストレージ機器のベンダーマニュアルにも記述があるので、機器マニュアルは必読

multipath-tools コマンド

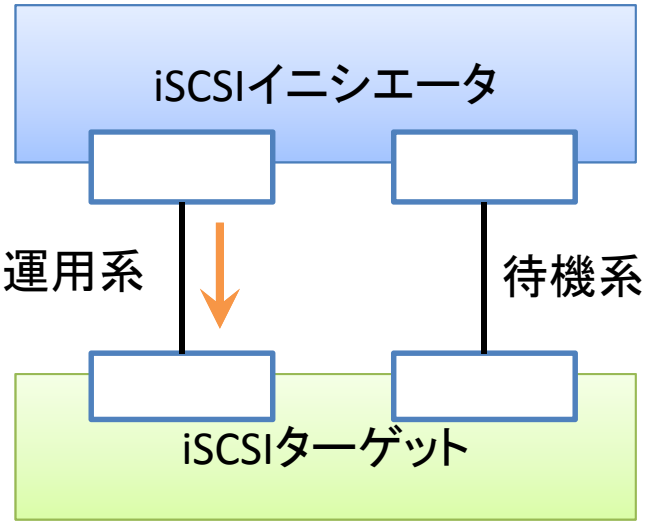
- multipath -l
- multipath -ll
  - Multipathのパスの状態を一覧表示する
- multipath -f <デバイス>
  - 指定されたマルチパスデバイスを削除
- multipath -F
  - 全てのマルチパスデバイスを削除
- multipath**d** -k
  - Multipathdのコンソール

multipath-tools FailOverの確認

```
[root@test01 ~]# multipath -ll
3600000e00d0000000003053800000000 dm-0 VENDER,PRODUCTS
[size=2.7T][features=1 queue_if_no_path][hwhandler=0][rw]
¥_ round-robin 0 [prio=1][active]
 ¥_ 9:0:0:0  sdb 8:16  [active][ready]
¥_ round-robin 0 [prio=1][enabled]
 ¥_ 10:0:0:0 sdg 8:96  [active][ready]
```

PathGroup
Path
PathGroup
Path

DM状態　Path状態



コンフィグ：

path_grouping_policy　failover

failover = 優先グループ毎に1つのパス

multibus = 1つの優先グループ内の全パスが有効

multipath-tools MultiBusの確認

```
[root@test01 ~]# multipath -ll
3600000e00d0000000003053800000000 dm-0 VENDER,PRODUCTS
[size=2.7T][features=1 queue_if_no_path][hwhandler=0][rw]
¥_ round-robin 0 [prio=1][active]
 ¥_ 5:0:0:0  sdb 8:16  [active][ready]
 ¥_ 6:0:0:0  sdc 8:32  [active][ready]
 ¥_ 7:0:0:0  sdd 8:48  [active][ready]
 ¥_ 8:0:0:0  sde 8:64  [active][ready]
```

PathGroup

Path

Path

Path

Path

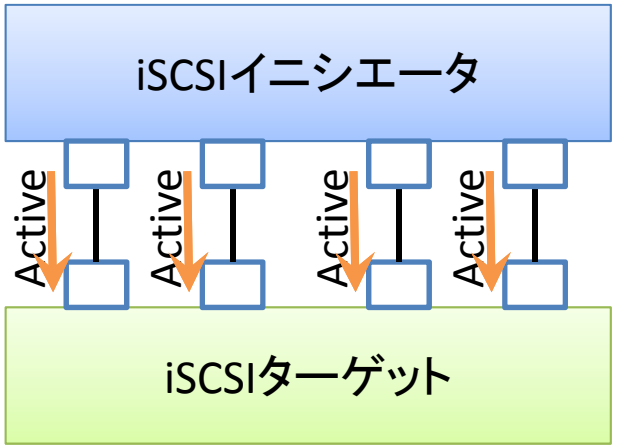
DM状態　　Path状態

コンフィグ：

path_grouping_policy　multibus

failover = 優先グループ毎に1つのパス

multibus = 1つの優先グループ内の全パスが有効



multipathd –k
実行例

```
multipathd> show paths
hcil     dev dev_t pri dm_st   chk_st  next_check
0:1:0:0  sda 8:0   1   [undef] [ready] [orphan]
9:0:0:0  sdb 8:16  1   [active][ready] X......... 2/20
10:0:0:0 sdg 8:96  1   [active][ready] XXXXXXX... 15/20
11:0:0:0 sdf 8:80  1   [active][ready] XX........ 5/20
12:0:0:0 sdh 8:112 1   [active][ready] XXXXXXXXX. 18/20

multipathd> show paths
hcil     dev dev_t pri dm_st   chk_st  next_check
0:1:0:0  sda 8:0   1   [undef] [ready] [orphan]
9:0:0:0  sdb 8:16  1   [active][ready] XXXXXXXXXX 20/20
10:0:0:0 sdg 8:96  1   [active][ready] XXXXXX.... 13/20
11:0:0:0 sdf 8:80  1   [active][ready] X......... 3/20
12:0:0:0 sdh 8:112 1   [active][ready] XXXXXXXX.. 16/20
```

抜ける場合には　Ctrl+D

ISCSI+MULTIPATH
運用上の注意点

JAPAN UNIX SOCIETY

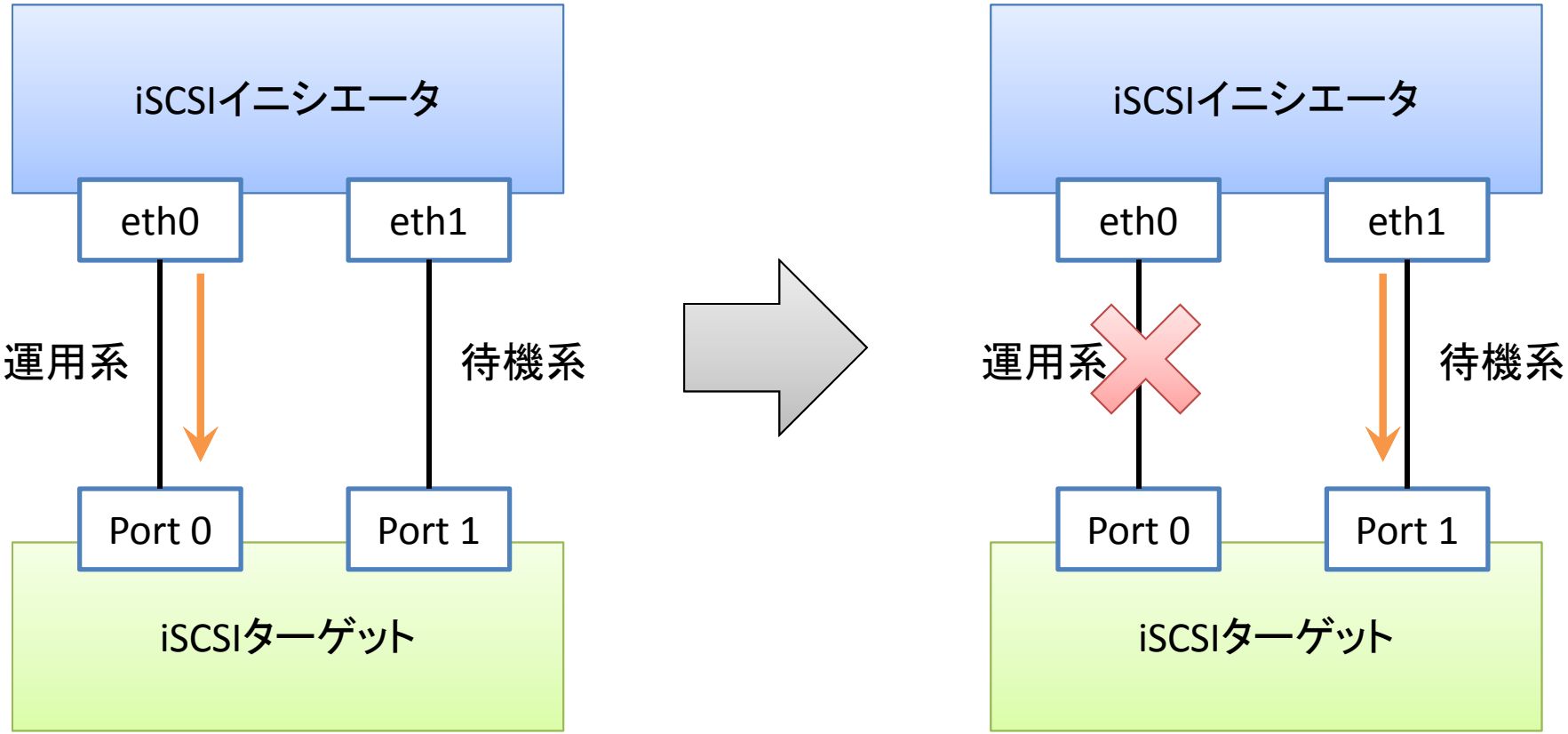
iSCSI+Multipathの注意点

- パスが切れた時、FailOverや切離しをしてくれるが、影響がないとは言えない

構成例

MultipathのFailover

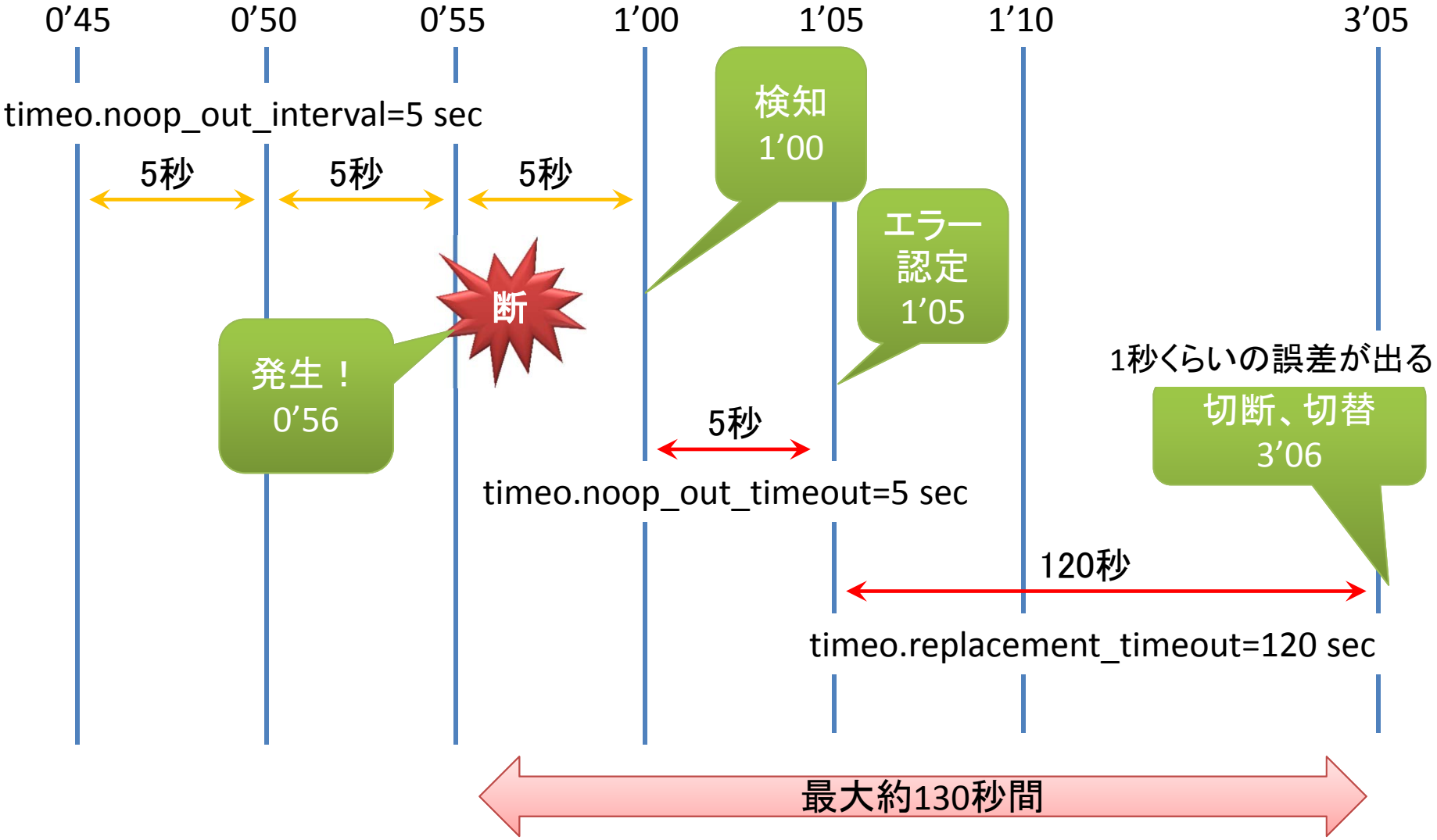
運用系で障害が発生すると、待機系に切り替わる



iscsid.confの確認 Default値

- node.conn[0].timeo.noop_out_interval = 5
  - Pingによる死活監視の間隔
- node.conn[0].timeo.noop_out_timeout = 5
  - PingによりエラーIと判断されるまでの時間
- node.session.timeo.replacement_timeout=120
  - エラーIと判断されてから、(Multipathから)切り離すまで

切替時間
Default値



切替時間

- 例えばケーブルが切れてから切り替わるまでの時間
  - 最大約130秒（Default値）
- アプリケーションには、30秒切れたらタイムアウト処理するのも多い
  - ちなみに、この間のiSCSIの処理は、キューに貯められ、フェールバックされます。

対応方法

- node.conn[0].timeo.noop_out_interval = 5
  - Pingによる死活監視の間隔
- node.conn[0].timeo.noop_out_timeout=5
  - Pingによりエラーと判断されるまでの時間
- node.session.timeo.replacement_timeout=120
  - エラーと判断されてから、（Multipathから）切り離すまで

これらを短くする。（自分の環境で行う場合には検証してください）
お勧めは、「node.session.timeo.replacement_timeout」の値の変更。

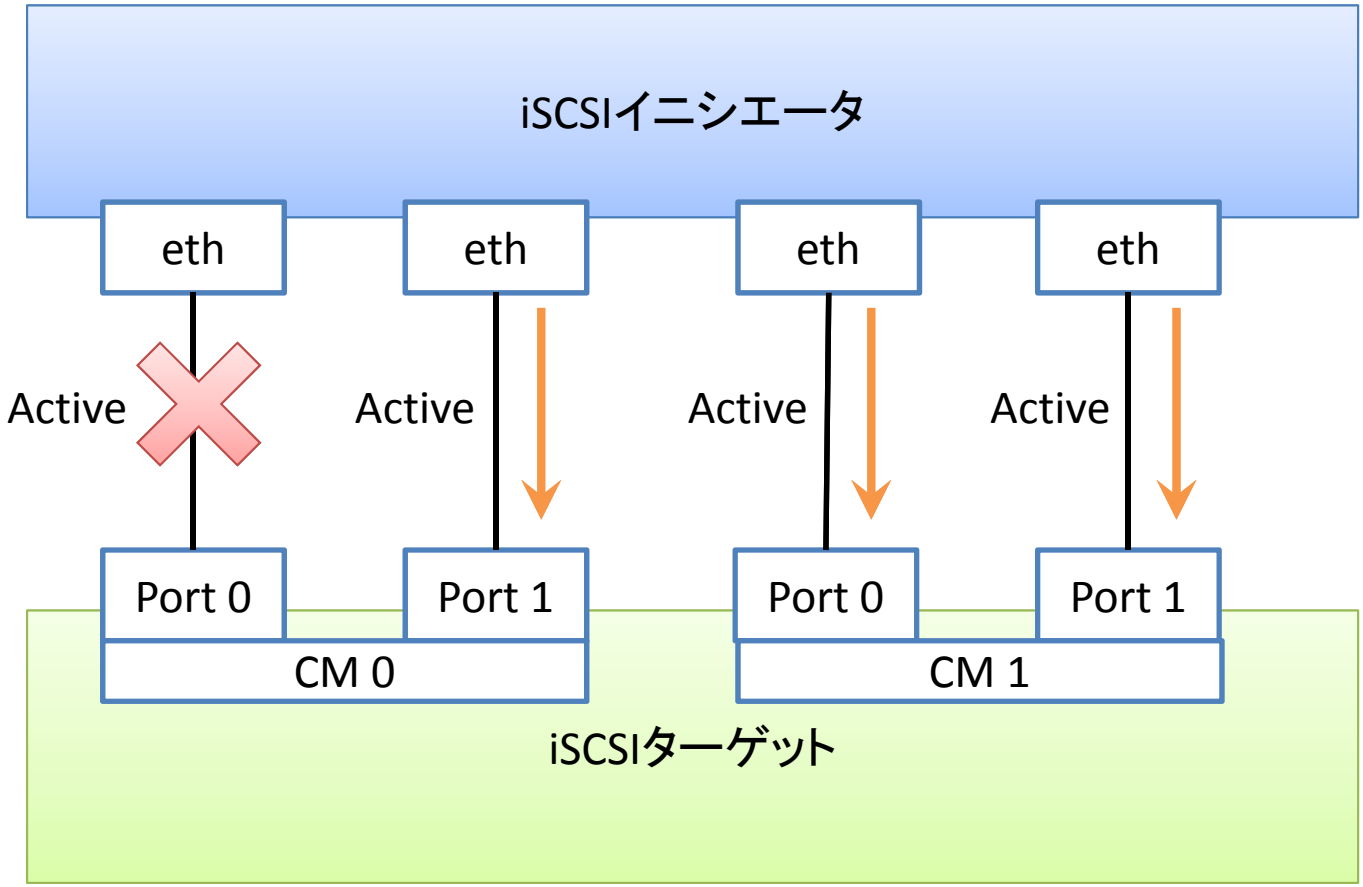
例えば、アプリのタイムアウトが30秒なら、それ以下になるように調整するなどの対応を推奨します。

対応方法 replacement_timeout

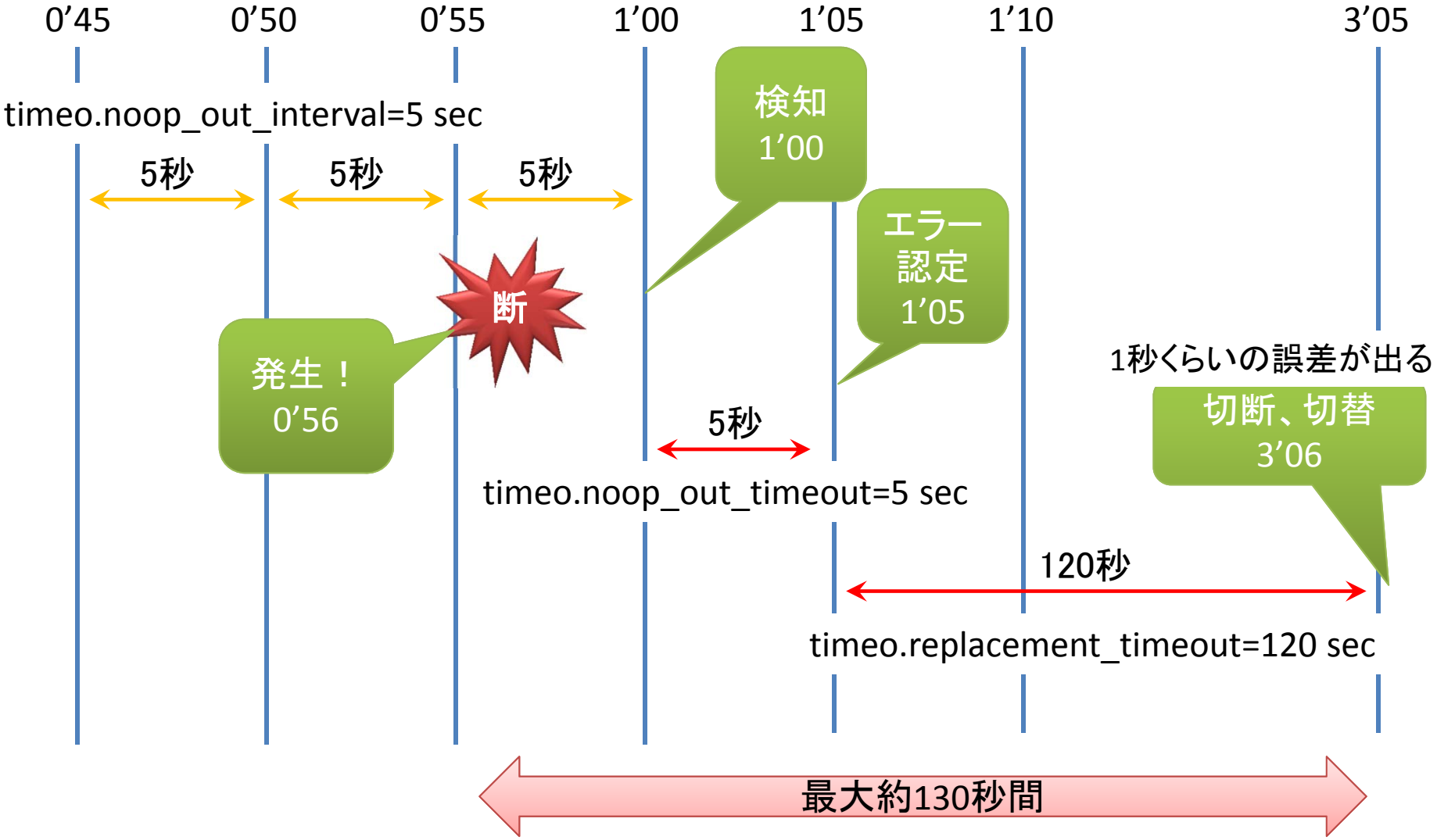
- open-iscsiのREADMEより
  - If the value is 0, IO will be failed immediately.
  - If the value is less than 0, IO will remain queued until the session is logged back in, or until the user runs the logout command.

- 結論
  - アプリのタイムアウトに合わせて適度に設定するのが妥当
  - ちょっと前のVersionでは、「0」設定がうまくいかなかったので、0以上の設定をお勧めします

構成例
MultipathのMultiBus



切替時間
Default値



結果は同様

- 4本中1本でも切れたら同じように全体に影響が出た

- 原因
  - IOの処理が細切れにされてしまうため
    - BIO(Block IO)ベースの問題
    - Kernel2.6.31以降で対応済みらしい（Requestベースに対応）

メンテナンス時の対応策

- iSCSIを収容しているSwitchのメンテナンスやケーブルのメンテナンスはどうすればいいの？
  - 明示的にログアウトする
    - iSCSIはセッションです。
    - ログアウトすればDMがよろしくやってくれます。

まとめ

- iSCSIは、実用レベルで利用可能だが、パフォーマンスや可用性の向上を目指して、日々Updateされている
  - 利用するストレージ機器とKernelのVersionを確認してからの利用を推奨する

ありがとうございました

Kernelリリースノート1

- Kernel 2.6.31
  - Requestベースのサポート
  - MappingTargetのバイナリサポートやFlushサポート
  - mpath: queue lengthとService time ロードバランシング追加（path_selector）
- Kernel 2.6.33
  - Requestベースのバイナリサポート
  - (Block関連)CFQ(Completely Fair Queuing; 完全公平型キューイング)強化
- Kernel 2.6.35
  - (MD関連) RAID0->RAID10 takeover,(0,4,5,10)
  - (Block関連) Block I/O controller (blkio)強化
- Kernel 2.6.36
  - MappingTargetのdiscardサポート (delay,linear,mpath,stripe)

Kernel リリースノート2

- Kernel 2.6.38
  - Improve significantly write throughput when writing to the origin with a snapshot on the same device
  - Improve sequential write throughput
  - dm-crypt: scale to multiple cpus
  - dm-crypt: add loop AES IV generator
  - RAID1: support discard
  - Skeleton for the DM target that will be the bridge from DM to MD (initially RAID456 and later RAID1). It provides a way to use device-mapper interfaces to the MD RAID456 driver

- Kernel 2.6.39
  - Add flakey target that it returns I/O errors periodically (commit)
  - stripe: implement merge method, performance improvement has been measured to be ~12-35% -- when a reasonable chunk_size is used (e.g. 64K) in conjunction with a stripe count that is a power of 2

Kernelリリースノート3

- Kernel 3.1
  - New iSCSI implementation
    - 今まで使用されてきたiSCSIの実装方式SCSTは、Linux-iSCSI.org SCSIターゲットを含めることで廃止されました。
  - flakey target: add corrupt_bio_byte feature , add drop_writes
  - Support the MD RAID1 personality through the dm-raid target
  - raid: Support metadata device