第15回FreeBSDワークショップ

佐藤　広生 <hrs@FreeBSD.org>

東京工業大学/ FreeBSD Project

2016/2/18

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

開催背景

- **日本国内の＊BSDユーザ活動を活発化させましょう**
  - 月1回、東京近辺で定期的な会合を。
  - 講演を聞くだけでなく、話を持ち寄って双方向に議論しましょう

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

本ワークショップの進行

- 19:00〜19:45　自己紹介＋話題にしたいトピック
- 19:45〜20:00　休憩
- 20:00〜20:50　ライトニングトーク
- 20:50〜21:00　最近のSA + α

意見は自由に発言ください！

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

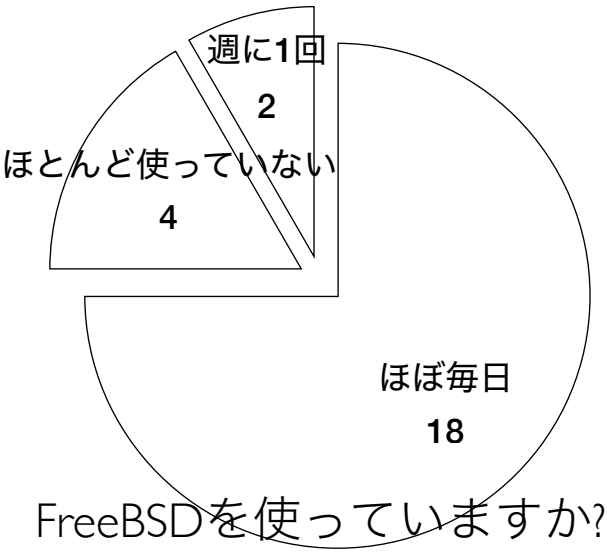
オーガナイザの自己紹介

- 名前：佐藤 広生
  - FreeBSD コアチームメンバ、リリースエンジニア(2006-)
  - FreeBSD Foundation 理事(2008-)
  - その他の＊BSD／オープンソース関連の活動いろいろ
  - 東京工業大学助教(2009-)

自己紹介タイム

- 名前（所属）
- 開発者 or 利用者
- 興味がある／話題にしたい内容

  をどうぞ



FreeBSDを使っていますか?

今回の出席者内訳：新規4名、再参加者18名

メモ

メモ

冗長ストレージデバイスの
疑似Active-Active化によるHA NFSサーバの構築

佐藤　広生 <hrs@FreeBSD.org>

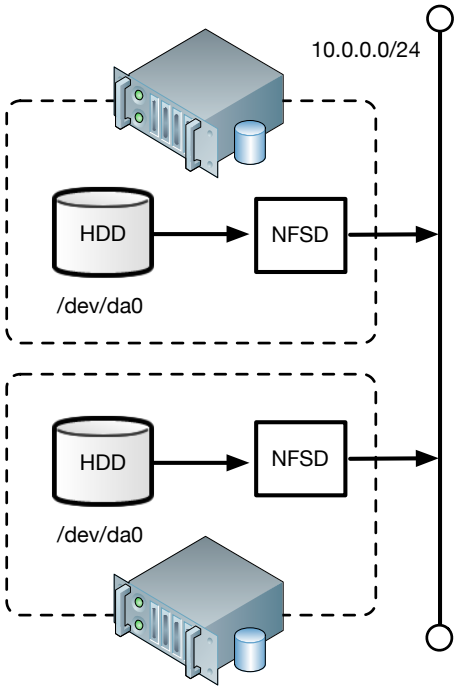
東京工業大学/ FreeBSD Project

2016/2/18

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

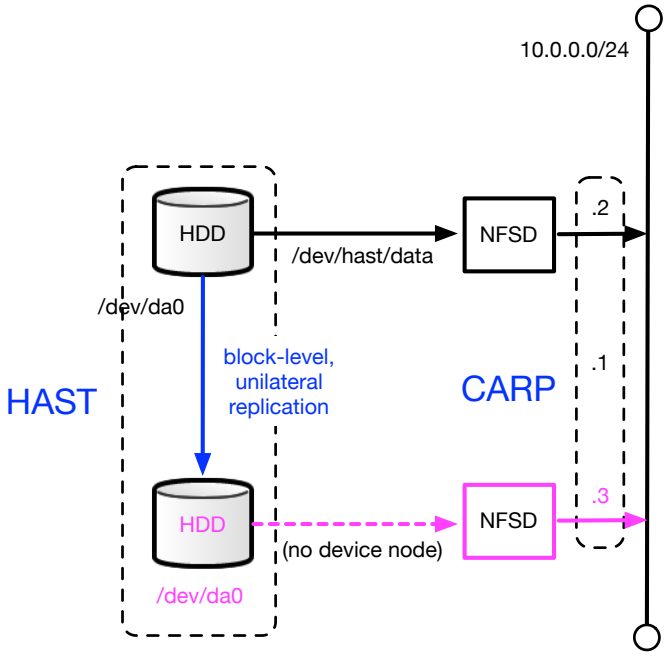
動機

- 許さんのページ
  - http://www.seirios.org/seirios/dokuwiki/doku.php?id=os:freebsd:hast



動機

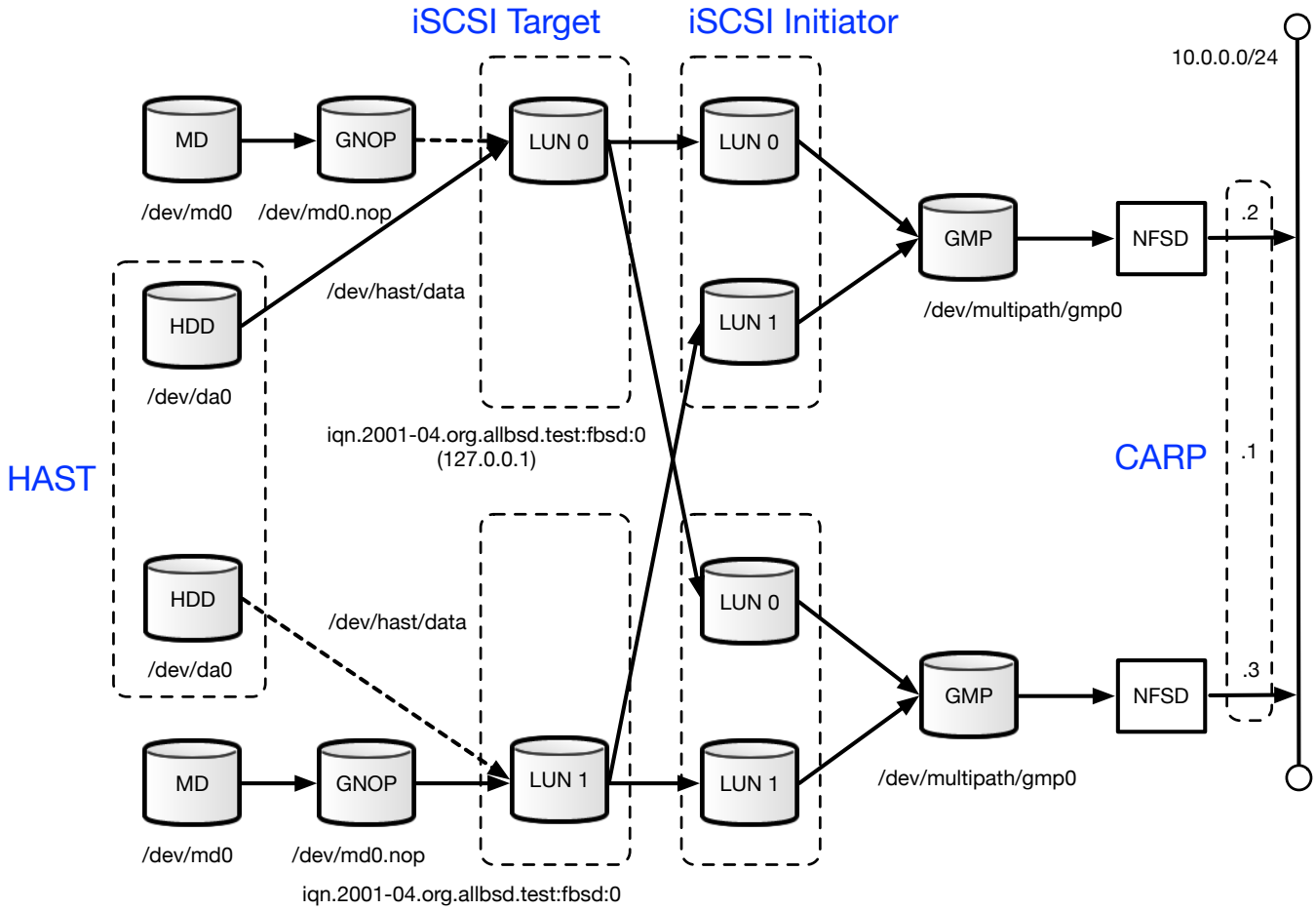
- ざっくりした説明
  - HASTを使って2台の
    バックエンドストレージを同期
  - primaryが/dev/hast/data を
    newfs & マウントしてnfsd で
    export
  - 仮想IPをCARPで割り当てて
    listen
  - primaryが死んだら、HASTを
    切り替えてマウントからexport
    までやり直し



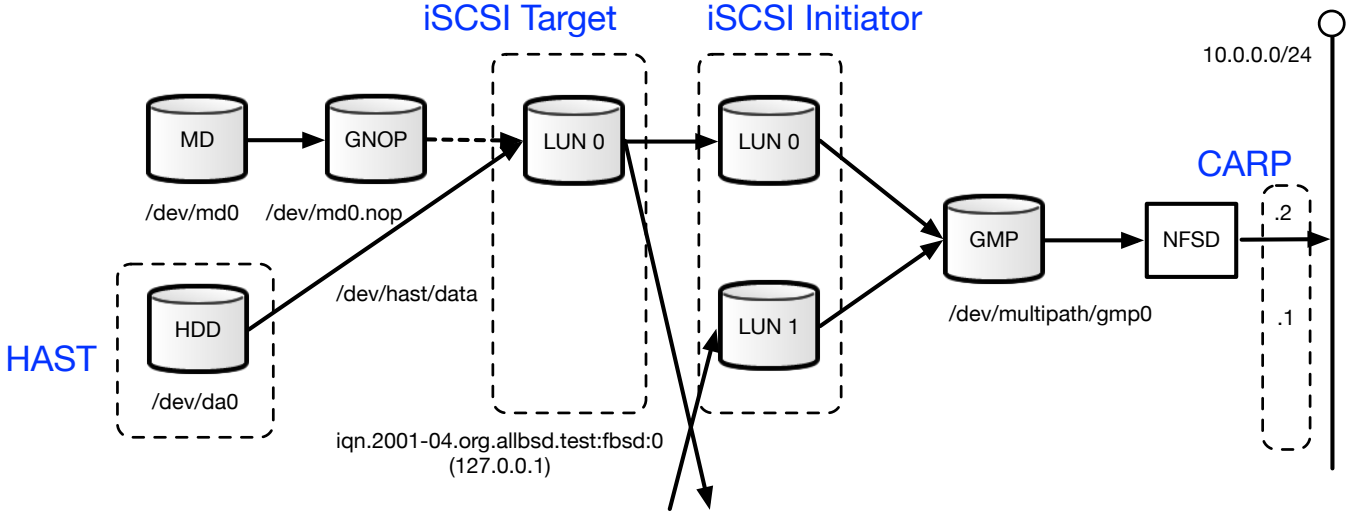
動機

- 問題点
  - ブロックストレージ、NFSDともにActive-Stanby
    - HASTは必ず手動で切り替える必要がある
    - CARPは自動で切り替えることができるが、
      この構成では手動で切り替えないとダメ
    - **切り替え同期は（やってできないことはないが）難しい**
- アプローチ
  - NFSDの（擬似）Active-Active化
    ＝ CARPの切り替えは、自動に任せる
    ＝ NFSDからはストレージのfailoverが見えないようにする
    （ストレージが全部死んだ時にはENXIOではなくEIOを返す）
  - まだいくつか未解決問題がありますが、まあとりあえず現状を。

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

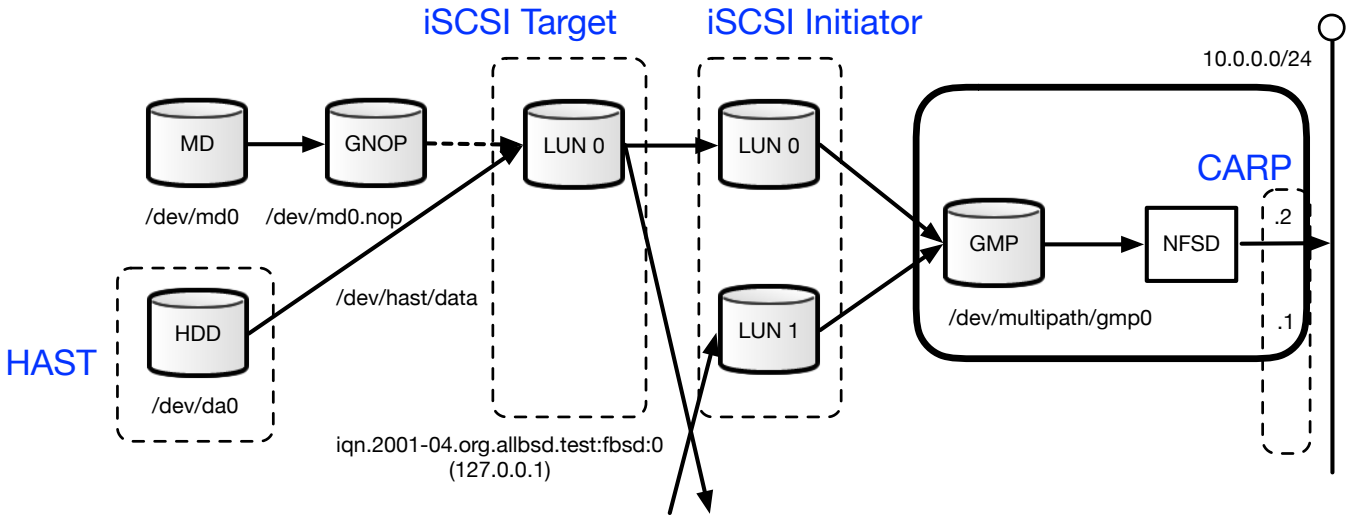


http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf



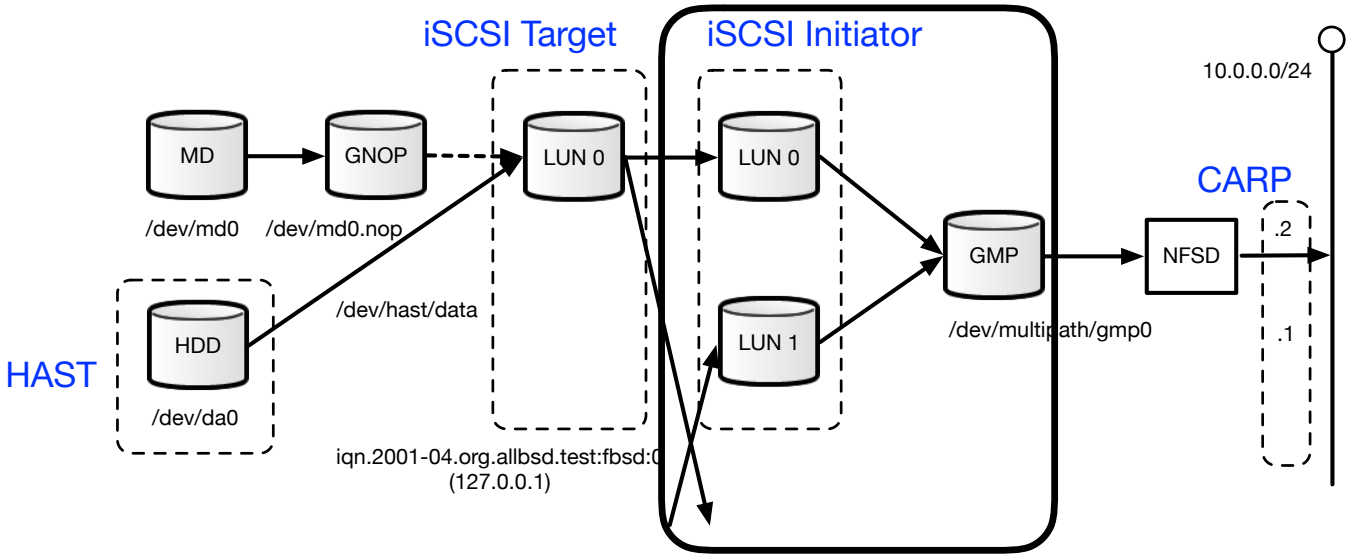
- 片方だけの構成図
- 右から説明します

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

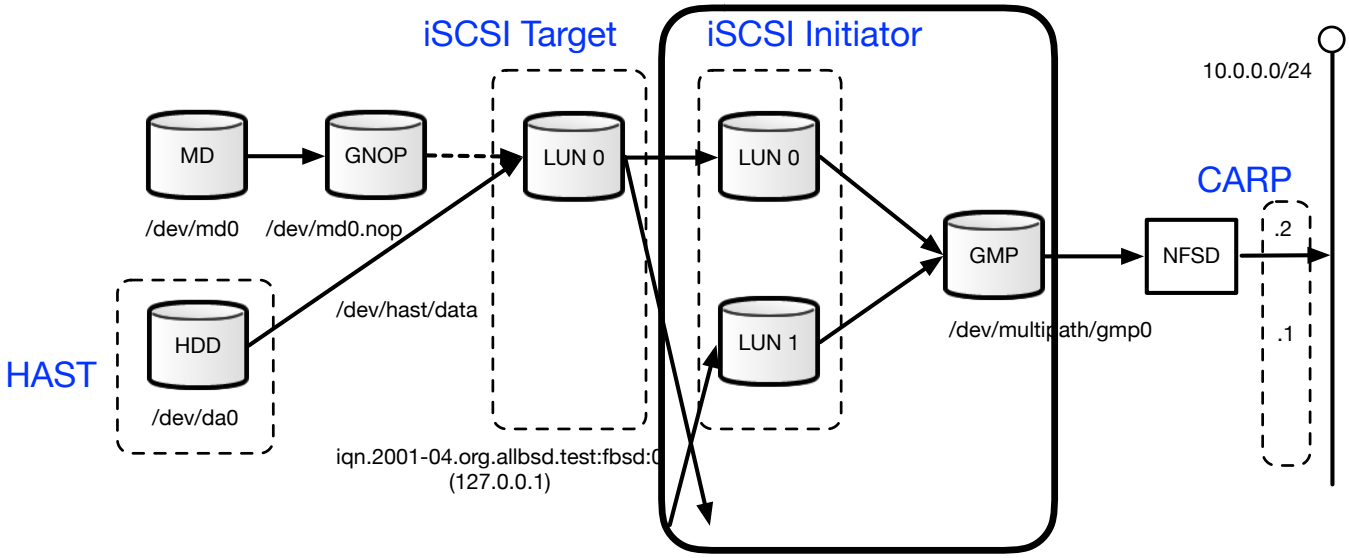


- /dev/multipath/gmp0をnewfs, マウントして使う
- 仮想IP部分（CARP）は特に変更なし

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

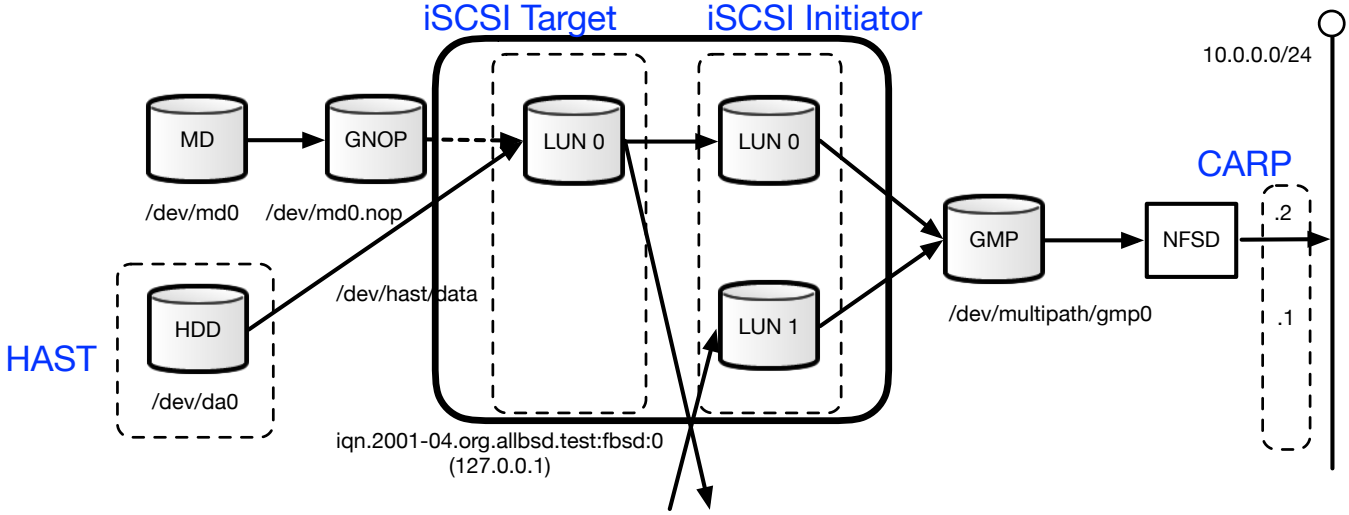


- GMULTIPATHとは？
  - 複数のI/O経路をまとめるGEOM provider
  - Active/Active, Active/Passive, Active/Read が選べる
  - 今回はActive/Passive で使う



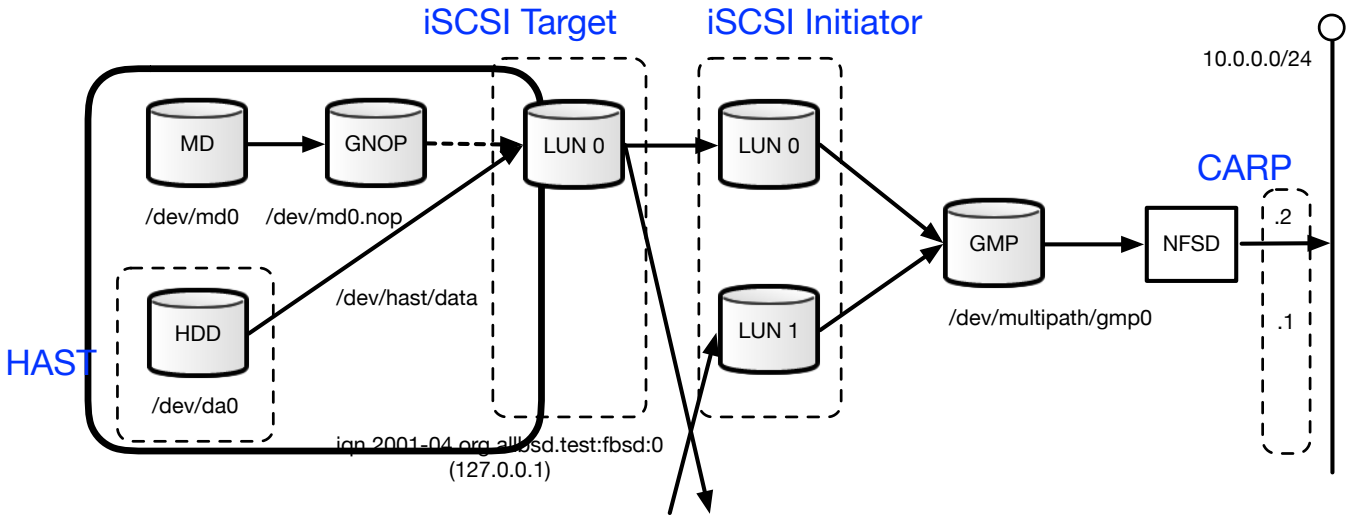
- GMULTIPATHを使って、2個のブロックデバイスを1個にする
  - LUN 0：HASTの片方、iSCSI経由
    LUN 1：HASTのもう片方、iSCSI経由
- 両方のHASTストレージをmultipath/gmp0に集約

http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf

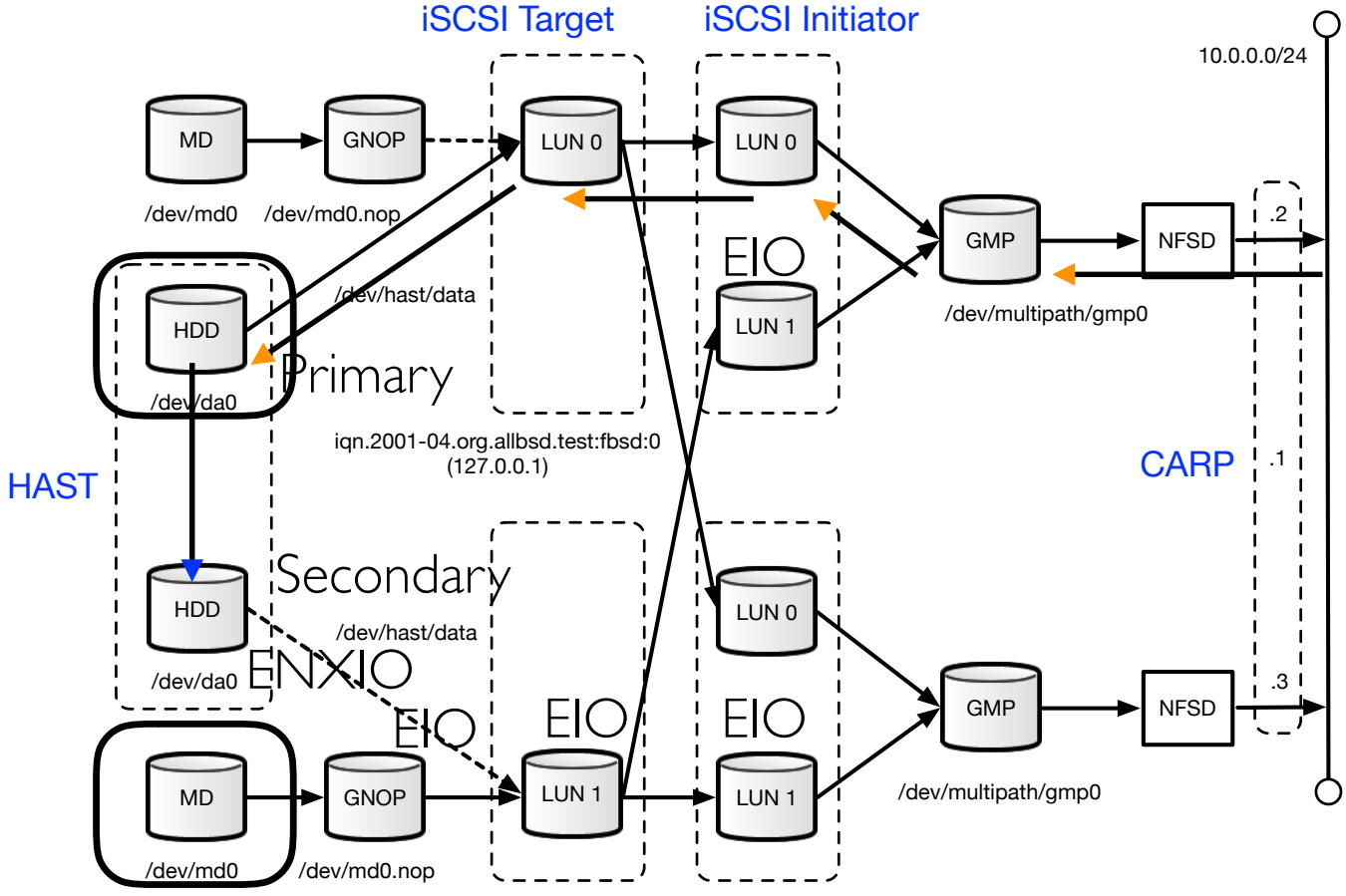


- iSCSIを使って、自分の/dev/hast/dataをexport
- 相手のhast/dataと、自分のhast/dataの両方を接続

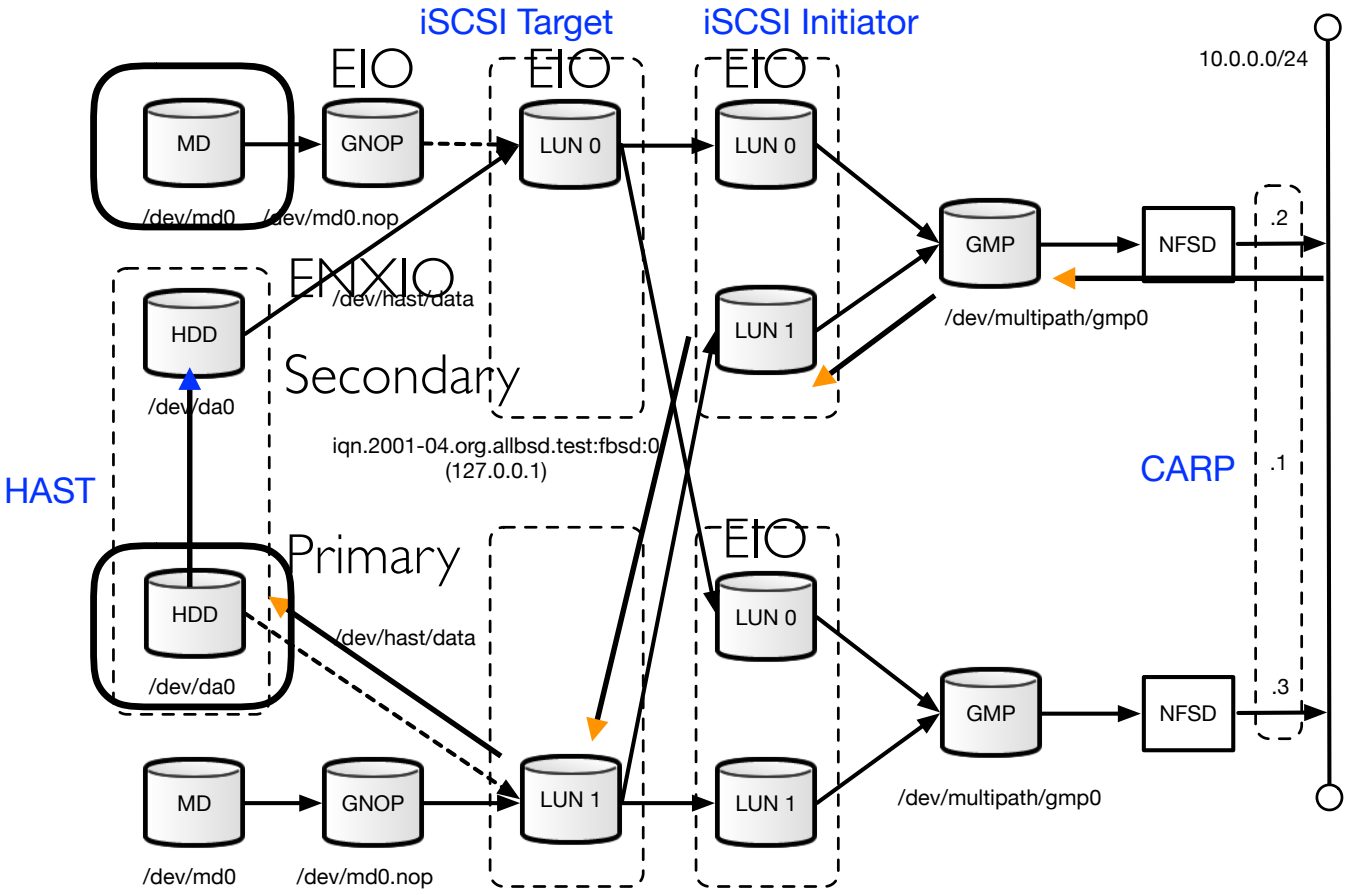
- 問題：どちらかのhast/dataは存在しないのでは？



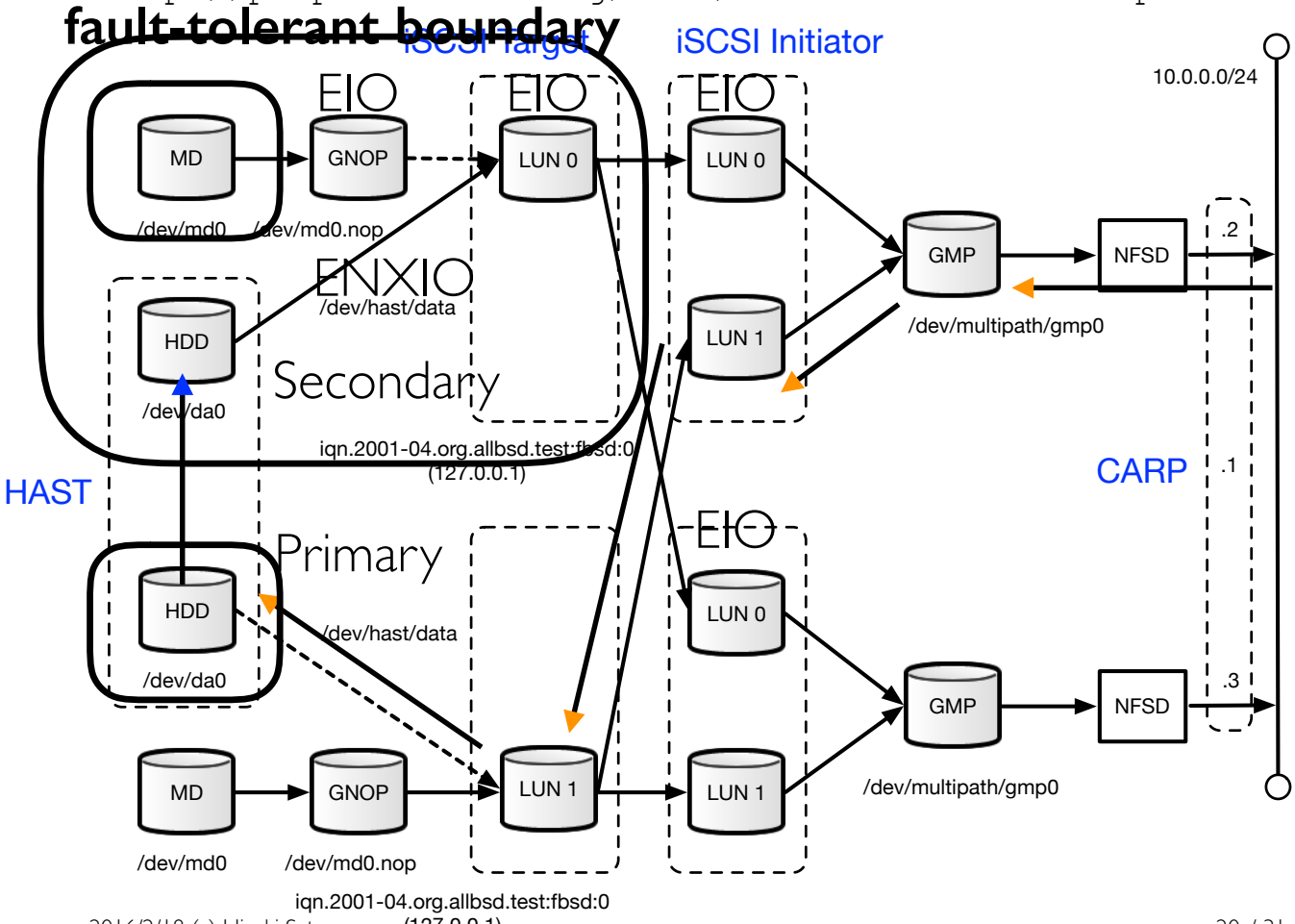
- 存在しないデバイスノードはiSCSIでexportできない！
  - 必ずエラーを返す仮想ストレージデバイスをGNOPで作る
- **primary: 「hast/data」**をLUN 0としてexport
  secondary: 「エラーデバイス」をLUN 0としてexport
  initiator側からは切り替え前後でLUN 0がすり替わって見える



http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf



http://people.allbsd.org/~hrs/sato-FBSDW20160218.pdf



障害モードと対処

- gmp0 より左側はActive-Stanbyなので、
  どちらか1系統を選択的に生かさないといけない

- HASTのprimary-secondary遷移で排他的に制御できる
  （gmultipathはENXIOにも対応しているので、デバイスが消えても大丈夫）

- HAST＋iSCSIの通信用ネットワークを別個に用意すべき
  - NFSサービスネットワークのダウンと独立させる
  - heartbeatを交換して、片方のダウン時の監視対応をする

設定

- 両方で ctld, iscsid, hastd を起動する
  - ctld は、LUNを入れ替えるために primary 用と secondary 用のconfigを作る
  - config の切り替えは、hastd から role 変更時に
    スクリプトを呼び出すことで自動化する

- roleとgmp0の状態
  - init - init or split-brain= gmp0はエラーを返すデバイス
  - primary - secondary =
    secondary - primary =
    どちらも gmp0 は正常なデバイス

設定

- /etc/rc.conf

```
hastd_enable="YES"
ctld_enable="YES"
iscsid_enable="YES"
iscsictl_enable="YES"
```

- /etc/iscsi.conf

```
box_1 {
  targetaddress = 127.0.0.1
  targetname = iqn.2001-04.localhost:0
}
box_2 {
  targetaddress = 10.0.0.3
  targetname = iqn.2001-04.3.0.0.10:0
}
```


設定

- /etc/ctl_primary.conf

```
portal-group pg0 {
        discovery-auth-group no-authentication
        listen 127.0.0.1
        listen 10.0.0.2
}

target iqn.localhost:0 {
        auth-group no-authentication
        portal-group pg0
        lun 0 {
#               path /dev/md99.nop
                path /dev/hast/data
                size 1g
        }
}
```

設定

- /etc/ctl_secondary.conf

```
portal-group pg0 {
        discovery-auth-group no-authentication
        listen 127.0.0.1
        listen 10.0.0.2
}

target iqn.localhost:0 {
        auth-group no-authentication
        portal-group pg0
        lun 0 {
                path /dev/md99.nop
#               path /dev/hast/data
                size 1g
        }
}
```


設定

- /etc/hast.conf

```
resource data {
  on box_1 {
    local /dev/da0
    remote tcp://10.0.0.3
  }
  on box_2 {
    local /dev/da0
    remote tcp://10.0.0.2
  }
  exec /etc/hast_event.sh
}
```

設定

- /etc/hast_event.sh

```
#!/bin/sh
if [ ! -r /dev/md99.nop ]; then
  # just after boot
  mdconfig -a -t swap -u 99 -s 1g
  gnop create -r 100 -w 100 -s 1g md99
  cp /etc/ctl_secondary.conf /etc/ctl.conf
  service ctld onerestart
fi
case $1 in
role)
  case $4 in
  primary|secondary)
    cp /etc/ctl_$4.conf /etc/ctl.conf
    service ctld onerestart
  ;;
  esac
esac
```


設定

- あとは gmultipath を使って da0 と da1 を束ねる
  - gmultipath create gmp0 da0 da1

- 起動時（HASTがinit状態の時）、
  gmp0 は「ディスクエラーが発生しているHDD」に見える。

- NFSクライアントからはActive/Activeに見える。
  - /dev/multipath/gmp0 を使っている限り、
    I/Oは box_1 or box_2 のどちらかへ排他的に
    ルーティングされる

課題

- gmp0上のUFSがasyncだと、バッファキャッシュによる
  書き込み遅延が発生する
  ＝本当にActive-ActiveでNFSクライアントからのアクセスを
  受け取る場合、UFSレベルで不整合が発生するような
  競合状態が存在する

- 2個の nfsd が /var/db/mounttab を上書きしてしまう

- 等々。

デモ

告知

- FreeBSDワークショップ（ほぼ月一回）
  （次回は4月）

- AsiaBSDCon 2016
  2016/3/10-13
  東京理科大学 森戸記念館
  飯田橋駅から徒歩5分、東京理科大学の施設

プログラム・参加登録受付は帰ったら開けます...