

19:10くらいから開始

第28回FreeBSDワークショップ

佐藤 広生 <hrs@FreeBSD.org>

東京工業大学/ FreeBSD Project

2017/4/28

2017/4/28 (c) Hiroki Sato

1

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
 - ▶ 困ったことや要望などはなるべく拾っていきます

本ワークショップの進行

- ▶ 19:00～19:30 自己紹介+話題にしたいトピック
- ▶ 19:30～21:00 ライトニングトーク

意見は自由に発言ください！

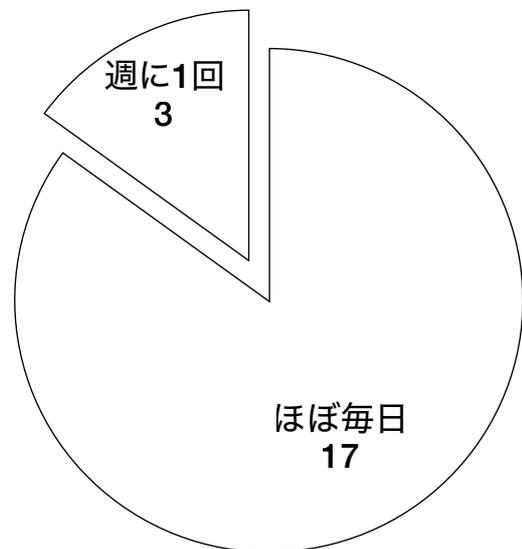
オーガナイザの自己紹介

- ▶ 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



今回の出席者内訳：新規1名、再参加者19名

メモ

メモ

FreeBSD 11.0Rのインストールと 小規模サーバ向けのストレージ構成

佐藤 広生 <hrs@FreeBSD.org>

東京工業大学/ FreeBSD Project

2017/4/28

2017/4/28 (c) Hiroki Sato

7 / 33

インストールしよう

- ▶ ISOイメージを取ってきて起動する
- ▶ メニューをポチポチ選択
- ▶ 終了

選択肢

- ▶ ディスクレイアウト
 - ▶ パーティションスキーム (MBR, GPT, ...)
 - ▶ ファイルシステム (UFS, ZFS,)
- ▶ オプション選択
 - ▶ MAC (強制アクセス制御) 設定

選択肢

- ▶ **UFS を使う**
 - ▶ MBR or GPT にする。2Tを超えたらGPT。
 - ▶ 冗長性はgmirrorで確保する
- ▶ **ZFS Root を使う**
 - ▶ GPTにする
- ▶ UEFIの場合はGPTしか使えない。
(起動はFATのEFIパーティションを使うのが普通)

選択肢

▶ UFS + gmirror

- ▶ UFSでの運用のまま、冗長性が確保できる
→ 昔ながらの管理者にもわかりやすい

▶ ZFS Root

- ▶ 64-bitでメモリが十分にあれば実用になる
- ▶ システム全体のスナップショットが取れる
→ ロールバック、バックアップが簡単

UFS + gmirror

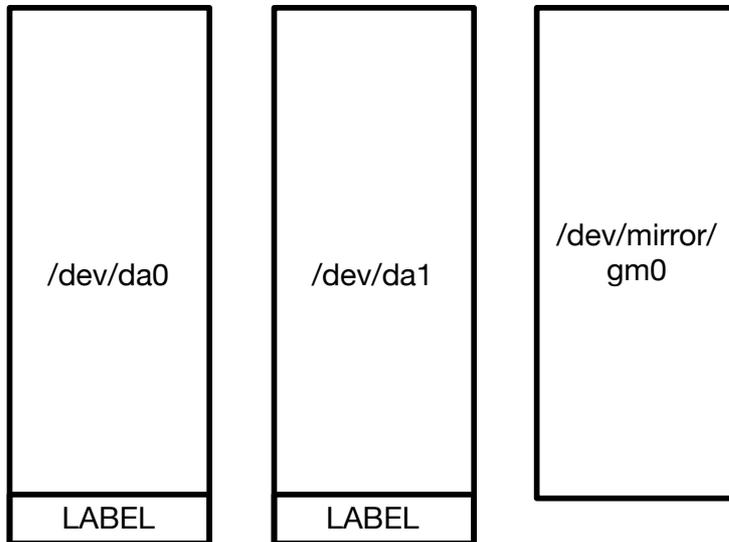
▶ GEOM_MIRROR とは？

- ▶ /dev/da0 + /dev/da1 → /dev/mirror/gm0
- ▶ セクタ単位でデータが複製される
- ▶ 読み込みはラウンドロビン、書き込みは2倍

▶ 作り方

- ▶ gmirror label gm0 da0 da1
- ▶ gm0の部分は自分で決めて良い

UFS + gmirror



- ▶ `da0`, `da1` の最終セクタに `gm0` の設定が書き込まれる
- ▶ カーネルが起動時に設定を検出して `/dev/mirror/gm0` が生える
- ▶ `gm0` は 1セクタ分だけ小さい

UFS + gmirror

- ▶ 問題： `da0` 単体で動いているシステムに `da1` を追加して `gm0` にするには？

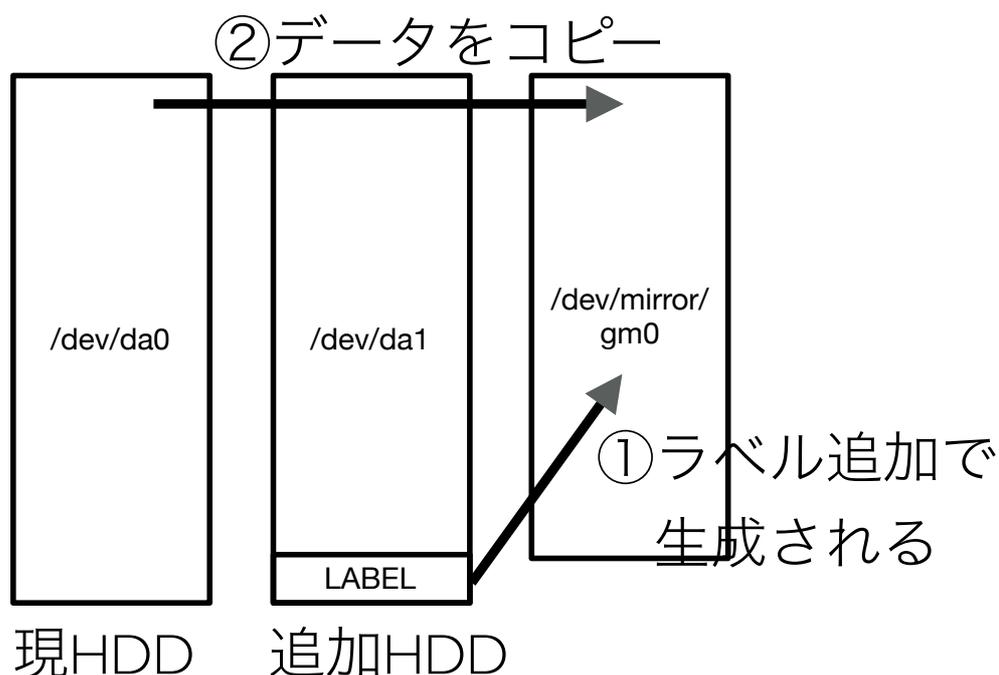
UFS + gmirror

- ▶ 問題：da0 単体で動いているシステムに da1 を追加して gm0 にはするには？
- ▶ da1 単体で gm0 を作る（ラベルをつける）
- ▶ da0 の内容を gm0 にコピー（dump & restore）
- ▶ gm0 の中の設定ファイルを gm0 に書き換え
- ▶ gm0 で起動
- ▶ da0 を gm0 に追加する（ラベルをつける）

2017/4/28 (c) Hiroki Sato

15

UFS + gmirror

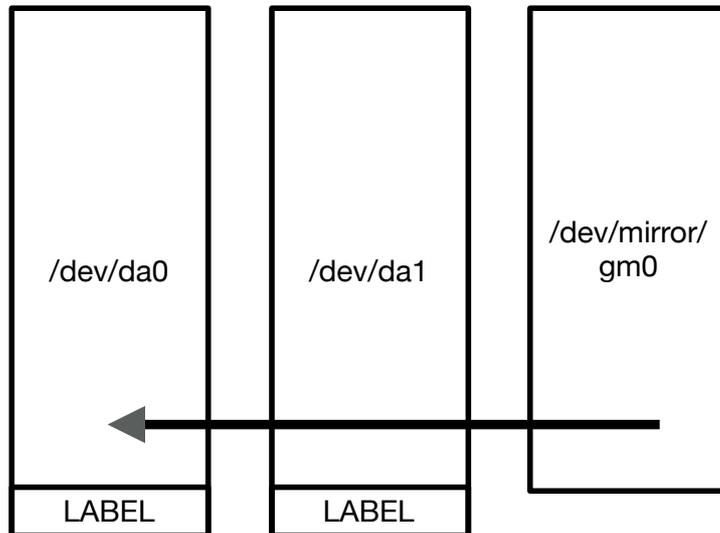


2017/4/28 (c) Hiroki Sato

16

UFS + gmirror

③ gm0で起動



④ da0にラベル追加でgm0→da0への複製処理開始

ZFS Root

▶ ZFS Root とは？

▶ ファイルシステム全体をZFSにする

```
% zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
zroot                                389M  16.9G   96K    /zroot
zroot/ROOT                           382M  16.9G   96K    none
zroot/ROOT/default                    452K  16.9G  381M    /
zroot/tmp                             4.73M  16.9G  4.73M  /tmp
zroot/usr                             384K  16.9G   96K    /usr
zroot/usr/home                        96K   16.9G   96K    /usr/home
zroot/usr/ports                       96K   16.9G   96K    /usr/ports
zroot/usr/src                         96K   16.9G   96K    /usr/src
zroot/var                             824K  16.9G   96K    /var
zroot/var/audit                       96K   16.9G   96K    /var/audit
zroot/var/crash                       96K   16.9G   96K    /var/crash
zroot/var/log                         148K  16.9G  148K    /var/log
zroot/var/mail                       100K  16.9G  100K    /var/mail
zroot/var/tmp                         96K   16.9G   96K    /var/tmp
```

ZFS Root

▶ ZFS Root とは？

- ▶ ファイルシステム全体をZFSにする
- ▶ 11.0Rからはインストーラで簡単にできる
- ▶ システム全体のスナップショットやロールバックが簡単
- ▶ BEs (Boot Environments) が便利

Boot Environments

- ▶ zpoolには、起動用データセットが指定可能
% zpool get bootfs <pool名>

```
% zpool get bootfs zroot
NAME      PROPERTY  VALUE                                     SOURCE
zroot    bootfs    zroot/ROOT/default                      local
```

- ▶ インストーラで入れると
ルートファイルシステム = zroot/ROOT/default
- ▶ この起動ファイルシステムを BE と呼ぶ

Boot Environments

- ▶ bootfs を書き換えると起動を変えることが可能
- ▶ 管理用のスクリプトが sysutils/beadm にある
- ▶ 典型的な応用
 - ▶ アップグレード
新しいBEを作って更新し、BEを切り替える
 - ▶ ベアメタル環境のjail化やその逆など

Boot Environments

```
% zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
zroot                389M  16.9G   96K    /zroot
zroot/ROOT           382M  16.9G   96K    none
zroot/ROOT/default 452K  16.9G  381M  /
zroot/tmp            4.73M 16.9G  4.73M  /tmp
zroot/usr            384K  16.9G   96K    /usr
zroot/usr/home       96K   16.9G   96K    /usr/home
zroot/usr/ports       96K   16.9G   96K    /usr/ports
zroot/usr/src         96K   16.9G   96K    /usr/src
zroot/var            824K  16.9G   96K    /var
zroot/var/audit       96K   16.9G   96K    /var/audit
zroot/var/crash       96K   16.9G   96K    /var/crash
zroot/var/log        148K  16.9G  148K    /var/log
zroot/var/mail        100K  16.9G  100K    /var/mail
zroot/var/tmp         96K   16.9G   96K    /var/tmp
```

beadm create

```
% beadm create -e default newbe
Created successfully
% zfs list -t snapshot,filesystem
NAME                                USED  AVAIL  REFER  MOUNTPOINT
zroot                                389M  16.9G   96K    /zroot
zroot/ROOT                          382M  16.9G   96K    none
zroot/ROOT/default                   452K  16.9G   381M   /
zroot/ROOT/default@2017-04-29-02:39:38 124K  -      381M  -
zroot/ROOT/newbe                      8K   16.9G   381M   /
zroot/tmp                            4.73M 16.9G  4.73M  /tmp
zroot/usr                           384K  16.9G   96K    /usr
zroot/usr/home                       96K  16.9G   96K    /usr/home
zroot/usr/ports                       96K  16.9G   96K    /usr/ports
zroot/usr/src                         96K  16.9G   96K    /usr/src
zroot/var                            824K  16.9G   96K    /var
zroot/var/audit                      96K  16.9G   96K    /var/audit
zroot/var/crash                      96K  16.9G   96K    /var/crash
zroot/var/log                        46K  16.9G   149K   /var/log
zroot/var/mail                       100K  16.9G   100K   /var/mail
zroot/var/tmp                        96K  16.9G   96K    /var/tmp
```

defaultのスナップショットを取ってnewbeにclone

beadm list

```
% beadm list
BE      Active Mountpoint  Space Created
default NR      /              382.0M 2017-04-28 19:22
newbe  -          -              132.0K 2017-04-29 02:39
```

BEリスト

N: 現在アクティブ

R: 次回起動時にアクティブ

beadm mount

```
# beadm mount newbe /mnt
Mounted successfully on '/mnt'
# mount
zroot/ROOT/default on / (zfs, local, noatime, nfsv4acls)
devfs on /dev (devfs, local, multilabel)
zroot/tmp on /tmp (zfs, local, noatime, nosuid, nfsv4acls)
zroot/usr/home on /usr/home (zfs, local, noatime, nfsv4acls)
zroot/usr/ports on /usr/ports (zfs, local, noatime, nosuid, nfsv4acls)
zroot/usr/src on /usr/src (zfs, local, noatime, nfsv4acls)
zroot/var/audit on /var/audit (zfs, local, noatime, noexec, nosuid, nfsv4ac
zroot/var/crash on /var/crash (zfs, local, noatime, noexec, nosuid, nfsv4ac
zroot/var/log on /var/log (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/mail on /var/mail (zfs, local, nfsv4acls)
zroot/var/tmp on /var/tmp (zfs, local, noatime, nosuid, nfsv4acls)
zroot on /zroot (zfs, local, noatime, nfsv4acls)
zroot/ROOT/newbe on /mnt (zfs, local, noatime, nfsv4acls)
```

マウントされる (内容は/defaultと同じ)

2017/4/28 (c) Hiroki Sato

25

更新する

```
# jail -c path=/mnt command=/bin/tcsh
root@:/ #
```

(ここで更新作業)

jail コマンドで /mnt に chroot し、
そこで更新作業などを行う。

2017/4/28 (c) Hiroki Sato

26

beadm activate

```
# beadm activate newbe ← BEの切り替え
Attempt to unmount boot environment 'newbe' mounted at '/mnt'
Gracefully unmounted boot environment 'newbe' from '/mnt' mount point
Activated successfully

# beadm list ← R が newbe に移動
BE      Active Mountpoint  Space Created
default N      /          308.0K 2017-04-28 19:22
newbe   R      -          382.1M 2017-04-29 02:39

# reboot ← 再起動
:
:
:

# beadm list
BE      Active Mountpoint  Space Created
default -      -          388.0K 2017-04-28 19:22
newbe  NR      /          382.1M 2017-04-29 02:39
```

BE と ローター



BE とローダー



Boot Environments

- ▶ beadmは、snapshot作成、clone作成、bootfs 書き換えを簡単に行うツール
- ▶ 問題：bootfs を書き換えてるだけなら手動でも行ける？

Boot Environments

- ▶ beadmは、snapshot作成、clone作成、bootfs 書き換えを簡単に行うツール
- ▶ 問題：bootfs を書き換えてるだけなら手動でも行ける？
- ▶ 答え：いろいろ落とし穴があるのでbeadm使いましょう

例：zfs clone した時に/entropy はどうすべき？

Boot Environments

- ▶ まとめ
 - ▶ BE = 起動用データセット
 - ▶ zpoolレベルではbootfsプロパティで選択
 - ▶ beadmでsnapshotやcloneも含めて管理可能
 - ▶ 作成したBEはmount + jail/chroot して操作すると便利
 - ▶ BEの一時的な切り替えはローダーからも可能
 - ▶ バックアップは単純にzfs send、リストアはzfs recvしてBE切り替え

まとめ

- ▶ サーバ向けの典型的な設定
 - ▶ UFS + gmirror
 - ▶ ZFS Root
- ▶ ZFS Root が許される環境なら、そちらの方が応用性は高い

告知

- ▶ FreeBSDワークショップ（ほぼ月一回）
 - ▶ 次回は5月19日
 - ▶ 発表の提案、具体的な解説の要望は随時歓迎