Bhyve Management Daemon

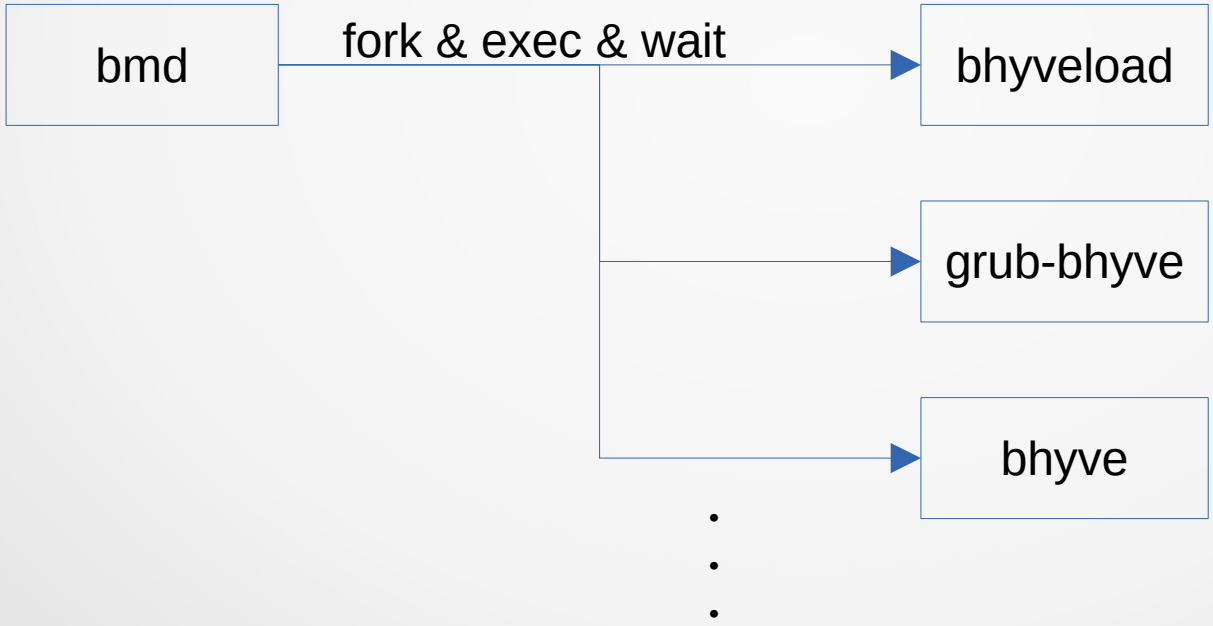
その後

2022 年 4 月 22 日

（株）創夢　内藤　祐一郎

bmd とは

- bhyve やローダのプロセスを管理するデーモン
- 設定ファイルに応じてプロセスを起動します



前回からの変更点

- 安定性強化
- qemu サポート（実験的）
- keyboard layout サポート
- UEFI VARS サポート
- boot kernel inspection

安定性強化

- procctl(2) の PROC_SPROTECT で OOM Killer に殺されないようにしました
- bmdctl との通信の完全非同期化
  - 処理の遅いクライアントがいてもデーモンの動作に支障がありません
  - また、タイムアウトを実装し通信が途絶したクライアントとの接続を自動的に切ります

qemu 対応

- 実験的サポートです
- 全ての設定が同じにできるわけではありません
  - qemu は vnc のディスプレイ番号を渡すインタフェースのため、`vnc port – 5900`で暫定的に計算します
  - boot はUEFI のみサポート
- FreeBSD arm64 が起動することは確認しました
  - 他は未確認です

keyboard layout 対応

- FreeBSD 14-Current の bhyve に vnc の keyboard layoutが指定できるようになりました
- bhyve が `-K <layout> ` を受け取るようになります
- 設定ファイルに `keymap` キーワードを追加しました

UEFI VARS 対応

- FreeBSD 14-Current の bhyve に
  EFI内部変数をファイルから読み書きする機能が
  追加されました
- デフォルトの値を $LOCALBASE/share/uefi-firmware/BHYVE_UEFI_VARS.fd から読み込み、各 VM 用に $LOCALBASE/var/cache/bmd の下に保存します
- インストール後に UEFI が HDD からブートするようになりました

boot kernel inspection

- NetBSD と OpenBSD のゲストに限り起動するカーネルを iso/disk イメージから自動的に調べる機能を実装しました
- 設定ファイルの loadcmd または installcmd に” auto” を渡すと調べた結果に基づき grub-bhyve に渡す boot コマンドを自動生成します

boot kernel inspection

- NetBSD の場合は起動ディスクに /netbsd カーネルファイルが存在するかどうかを調べます
- OpenBSD の iso イメージの場合、” /x.y/amd64/bsd.rd” ファイルの有無を調べます（ x.y はバージョン番号）
- OpenBSD の disk イメージの場合、” /bsd.upgrade” または ” /bsd” の有無を調べます

困ったこと

- 開発していて困ったことをまとめます
  1. 仮想マシン破棄後に vmm の内部状態が完全にクリアされるタイミングがユーザランドから分からない
  2. OpenBSD の二つ目以降の disklabel パーティションが読めない（先頭のみ読める）

vmm 破棄タイミング

- service bmd restart 時に以下を行います
  1. 全仮想マシン停止（acpi shutdown）
  2. bmd 停止
  3. bmd 起動
  4. 仮想マシン起動
- １～４が高速に実行されるため、カーネル内部の破棄処理が終わる前に同じ仮想マシンを作成しようとしてエラーになります
- /dev/vmm/<マシン名> の削除を監視してもダメでした

vmm 破棄タイミング

- sys/amd64/vmm/vmm_dev.c: sysctl_vmm_destroy() 内で設定しているコールバック関数で破棄処理が走ります

sys/amd64/vmm/vmm_dev.c:

```
1076        SLIST_FOREACH(dsc, &sc->devmem, link) {
1077                KASSERT(dsc->cdev != NULL, ("devmem cdev already destroyed"));
1078                destroy_dev_sched_cb(dsc->cdev, devmem_destroy, dsc);
1079        }
1080        destroy_dev_sched_cb(cdev, vmmdev_destroy, sc);
```

- おそらくこの vmmdev_destory 関数が device ファイルの削除後に呼び出されるのだろうと推測しています

OpenBSD の disklabel

- 話を簡単にするために、
  MBR で disklabel を使った場合について説明します
- opentest という仮想マシン用のイメージを zvol 上に作成します
- そこに OpenBSD をインストールすると次ページような MBR レイアウトになります

OpenBSD の disklabel

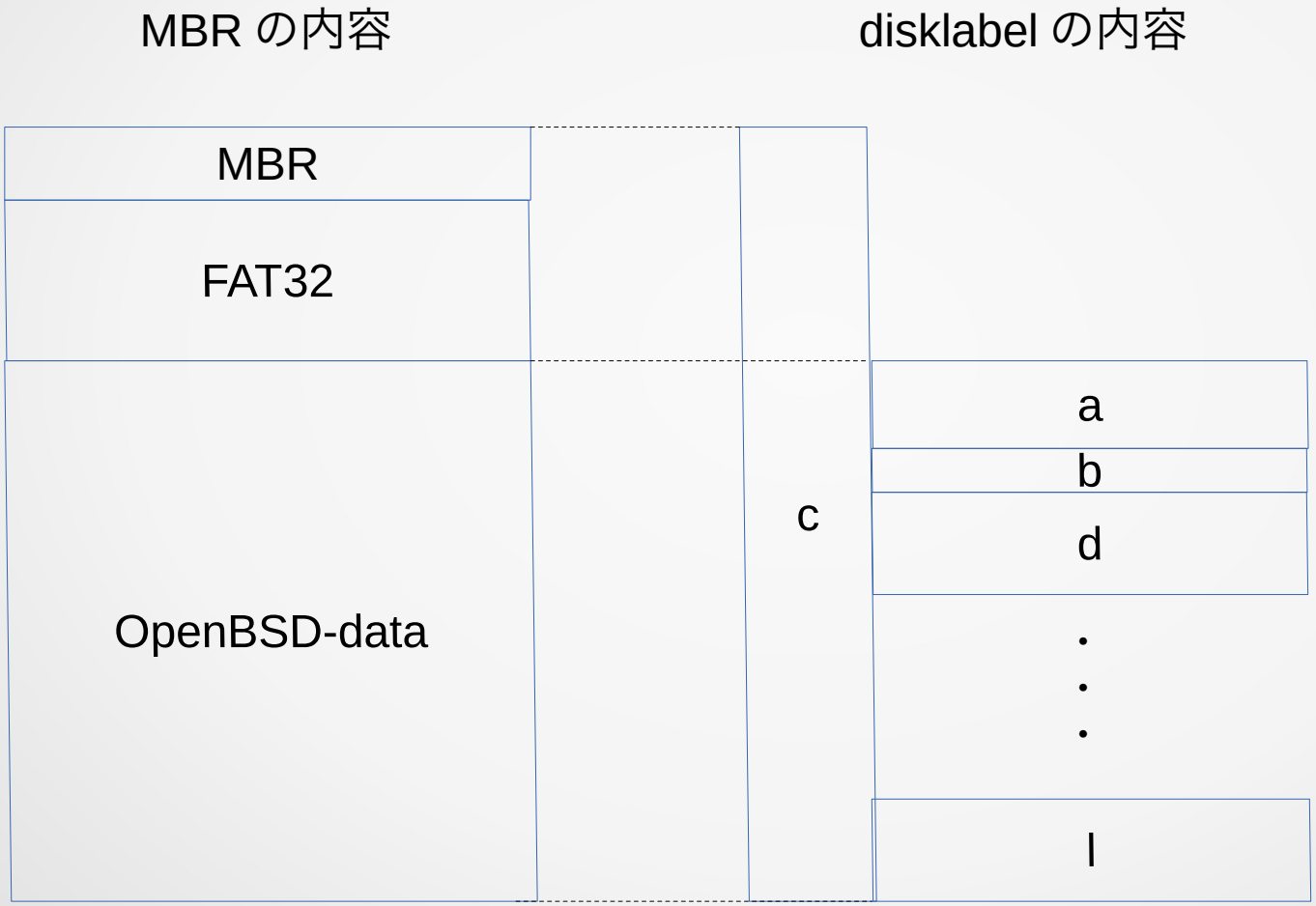
```
# gpart show /dev/zvol/zpool/images/opentest
=>       63  41942977  zvol/zpool/images/opentest  MBR  (20G)
         63         1                              - free -  (512B)
         64   2097152                              1  fat32  (1.0G)
    2097216  39845824                              4  openbsd-data  (19G)
```

一致

```
# disklabel /dev/zvol/zpool/images/opentests4
# /dev/zvol/zpool/images/opentests4:
16 partitions:
#          size     offset    fstype   [fsize bsize bps/cpg]
  a:    1195392    2097216    4.2BSD        0     0  9339
  b:    1940260    3292608      swap
  c:   41943040          0    unused        0     0     # "raw" part,
don't edit
  d:    1666848    5232896    4.2BSD        0     0 12919
  e:    2473184    6899744    4.2BSD        0     0 12960
  f:    4848416    9372928    4.2BSD        0     0 12960
  g:    1319328   14221344    4.2BSD        0     0 10225
  h:    4761760   15540672    4.2BSD        0     0 12960
  i:    2097152         64     MSDOS
  j:    3017664   20302432    4.2BSD        0     0 12960
  k:   11196320   23320096    4.2BSD        0     0 12960
  l:    7426624   34516416    4.2BSD        0     0 12960
```

合計

OpenBSD の disklabel



OpenBSD の disklabel

- FreeBSD から zvol の opentests4 をマウントすると a パーティションがマウントできる
- opentests4a ではマウントできない
- 同様に opentests4[d-l] もマウントできない
- boot kernel inspection は先頭パーティションのみ探索可能

最後に

- boot kernel inspection は便利です
- Linux 版の作成には課題が多いです
  - ファイルシステムのマウント（ fusefs を使うのか？）
  - grub.conf の解析が必要かも
  - そもそも grub-bhyve が xfs 未サポート