

ハイパーバイザの作り方～ちゃんと理解する仮想化技術～ 第18回 FreeBSD 10.0-RELEASE 公開記念 10.0-RELEASE で学ぶ bhyve の使い方

はじめに

2014年1月20日に bhyve がマージされた FreeBSD10.0 がついにリリースされました。今回はこれを記念し、仮想マシンのネットワークデバイスについて解説する予定を変更し、bhyve の使い方をおさらいします。

bhyve とは

bhyve は NetApp により開発され 2011 年に発表された、新しいハイパーバイザです。設計は Linux KVM に似ており、大雑把に説明すれば KVM の FreeBSD 版と言えます。但し、KVM ではユーザランドプログラムに既存のエミュレータである QEMU を流用しているため既存 OS との高い互換性が得られています。しかし、コードの見通しが悪くなってしまっているのが難点として挙げられます。一方、bhyve では独自に一から実装したユーザランドにより、シンプルで見通しのよいコードを保っています。

bhyve の動作環境

bhyve を試すには、Intel VT-x と EPT (Extended Page Tables) をサポートした CPU が必要です。今のところ AMD の CPU には対応していません^{*1}。

どのモデルの CPU が EPT をサポートしているかは ark.intel.com で調べられます。簡単な目安としては、Core i3, i5, i7 ならばほぼ全ての CPU が対応しています。Celeron や Pentium などの廉価版 CPU でも EPT 対応モデルが一部存在しています。実機にインストールするのが最も確実ですが、最近の VMware (VMware Player ・ VMware Workstation ・ VMware Fusion) ならば Nested VM に対応しているため仮想マシン上で試すことも出来ます。

ただし、FreeBSD 以外のゲスト OS を動作させるためには、上述の条件を満たす環境でも以下のような問題が確認されています。

^{*1} svm ブランチで開発が進められています：http://svnweb.freebsd.org/base/projects/bhyve_svm/sys/?sortby=date&view=log

- VMware による Nested VM 環境へ Linux をゲスト OS として起動しようとする、タイマ周りの問題で Linux カーネルがフリーズする
- 古い Core i シリーズ (Nehalem) は unrestricted guest (注: VM でリアルモードをサポートする機能) をサポートしないため、grub2-bhyve (後述) で Linux や OpenBSD カーネルを起動出来ない
- 一部のバージョンの Linux カーネルと一部の Intel CPU の組み合わせで非サポート MSR のアクセスによるエラーを起こして bhyve が異常終了する (後述)

また、bhyve を実行するには amd64 版の FreeBSD を使用する必要があります。32bit 版ではサポートされていません。

bhyve が提供する機能

現状の bhyve では幾つかのゲスト OS をロードし実行するのに最低限の機能のみが実装されています。

ハードディスクコントローラとしては AHCI コントローラのエミュレーションと、準仮想化ドライバである virtio-blk をサポートしています。NIC コントローラは、準仮想化ドライバである virtio-net をサポートしており、Intel e1000 のような標準的なデバイスエミュレーションはサポートしていません。virtio は多くの場合、標準的なデバイスエミュレーションと比較して高い性能が得られますが、ゲスト OS に virtio ドライバをインストールされており起動時に使えるように設定する必要があります。

システムコンソールとしては PCI 接続の 16550 互換シリアルポートをサポートしており、標準的なビデオデバイスはサポートしていません。また、X11 を起動して GUI 環境を表示する事はできません。

bhyve がエミュレート可能なデバイスは上述のものですが、Intel VT-d を用いて実機上の PCI・PCI Express デバイスをゲストマシンへパススルー接続できます。

このほか割り込みコントローラのエミュレーション (Local APIC、IO-APIC) や、タイマデバイスのエミュレーション、ハードウェア構成をゲスト OS へ伝えるのに必要な APIC などをサポートしています。

また、BIOS や UEFI などのファームウェアをサポートしていないため、ディスクイメージからブートローダをロードしてゲスト OS を起動する事ができません。このためにハイパーバイザの機能としてゲスト OS をロードしゲストマシンを初期化する OS ロードが実装されています。

bhyve の構成

bhyve は CPU に対して VT-x 命令を発行するなどハードウェアに近い処理を行うカーネルモジュール (vmm.ko) と、ユーザランドにおいてユーザインタフェースを提供しハードウェアエミュレーションを行う VM 実行プログラム (/usr/sbin/bhyve) の二つからなります。

前述のとおり、bhyve は BIOS や UEFI などのファームウェアをサポートしておらず、ディスクイメージ上のブートローダを実行できません。このため、ゲストカーネルをロードして起動可能な状態に初期化するゲスト OS ロード (/usr/sbin/bhyveload) が付属します。bhyve ではディスクイメージ上のブート

ローダを実行できないため、ゲストカーネルをロードして起動可能な状態に初期化するゲスト OS ローダ (/usr/sbin/bhyveload) が付属します。

/usr/sbin/bhyveload は FreeBSD ブートローダを FreeBSD 上で実行可能なプログラムに改変し、ゲストマシンのディスクイメージからカーネルを読み込んでゲストメモリ空間へ展開するようにしたものです。

/usr/sbin/bhyveload を実行すると、FreeBSD のブート時に表示されるのと同じメニューが表示されます。

このため、一見するとゲストマシンの実行が開始されたように見えます。しかし、これはホスト OS で /usr/sbin/bhyveload が出力している画面で、ゲストマシンは起動していません。また、VM インスタンスの削除などを行うための VM インスタンス管理ツールとして /usr/sbin/bhyvectl が提供されています。

これらのユーザランドのプログラム群は、VM 管理用のライブラリ (libvmmapi) を通して vmm.ko が提供するデバイスに対して mmap や ioctl を発行してゲストマシンの初期化や実行を行います。全体図を図 1 に示します。

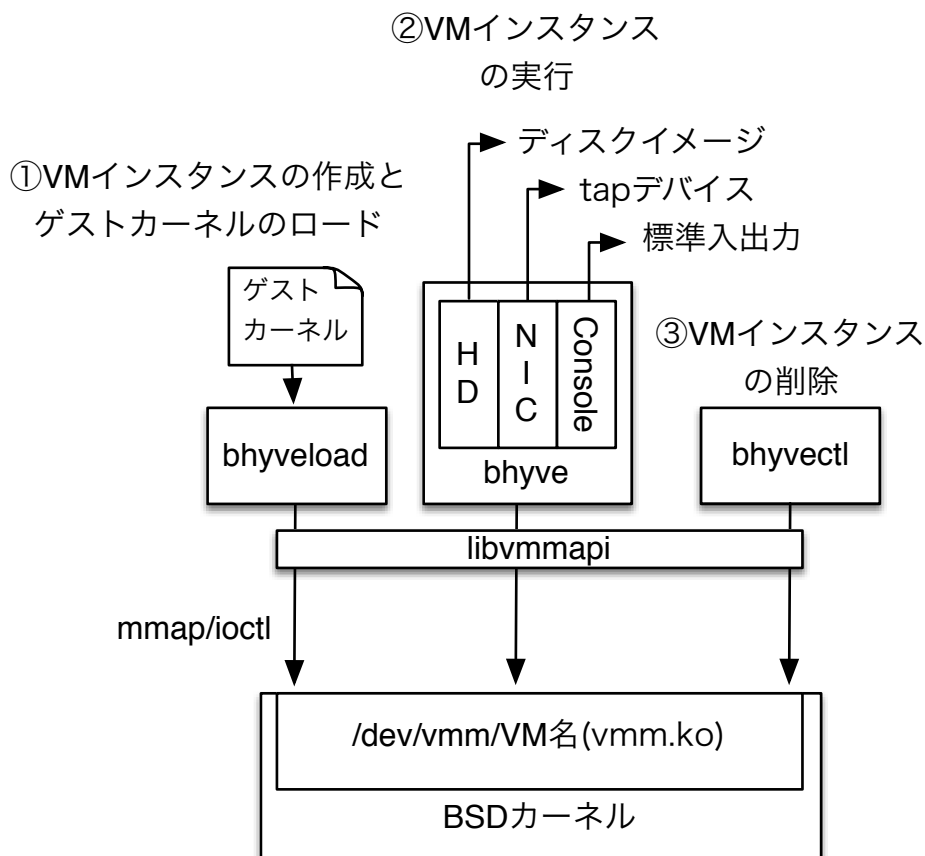


図 1 bhyve の構成

grub2-bhyve

前述の `/usr/sbin/bhyveload` では、FreeBSD 以外のゲスト OS を起動させられません。これを解決するために、GRUB2 を FreeBSD 上で実行可能なプログラムに改変した `grub2-bhyve` が提供されています。`grub2-bhyve` は `/usr/sbin/bhyveload` と異なり汎用 OS ロードであるため、GRUB2 がサポートする各種 OS を起動させる事ができます。

これまでに動作が確認されている OS は以下の 3 つです。 * Linux * OpenBSD * FreeBSD (`grub2-bhyve` 経由)

`/usr/sbin/bhyve` や `/usr/sbin/bhyveload` は FreeBSD base に付属するプログラムであるのに対し、`grub2-bhyve` は Ports · pkgng 経由で提供されるパッケージとなっています。

vmrc

`vmrc` は `bhyve-script` を更に拡張したもので、`bhyve` の他に `jail` や `QEMU` もサポートします。今回は `bhyve-script` を用いた VM の構築方法について紹介しますが、今後はこちらが主流のツールになる可能性があります。うです。

bhyve-script

QEMU/KVM のコマンド引数はユーザにとって分かりづらく、KVM ベースの仮想マシンの構築には `libvirt` + `virsh` などのフロントエンドが多く用いられています。

`libvirt` は今のところ `bhyve` をサポートしていませんが、その代わりになるような簡易的なシェルスクリプトが `bhyve.org` から提供されています。ダウンロードページへの URL はこちらです : <http://bhyve.org/tools/>

各種ゲスト OS のインストール方法

以下に、FreeBSD 10.0-RELEASE における各種ゲスト OS のインストール方法を示します。なお、全てのゲスト OS は amd64 (`x86_64`) 版であり、32bit 版はサポートされていません。

事前に必要な作業

`grub2-bhyve` と `bhyve-script` をインストールします。

```
# pkg install grub2-bhyve tmux
# fetch http://bhyve.org/bhyve-script.tar
# tar -xvf bhyve-script.tar
```

Ubuntu 13.10 の場合

```
# cd bhyve-script
```

```
# cp vm0 ubuntu0
```

これから作る VM の名前に vm0 をコピー（末尾は数値で、他の VM と重複しない値でなければならない）

```
# vi ubuntu0
```

NIC= " em0 " をお使いの NIC 名に

VCPUS= " 1 " を任意の vCPU 数に

VMRAM=1024 " を任意のメモリサイズに

VMOS= " freebsd " を " linux " に

VMOSVER= " 9.2-RELEASE " を " ubuntu13.10 " に

DEVSIZE= " 2G " を任意のディスクサイズに

それぞれ変更。

```
# sh ubuntu0 iso
```

ISO ファイルがダウンロードされ、ブートローダが起動する。

ネットワークの設定画面で DHCP が失敗するため、「Do not configure the network at this time(ネットワークの設定をしない)」を選択。

パーティション作成画面では「 Guided - use entire disk (LVMを使わない)」を選択。

```
# sh ubuntu0 start
```

HD イメージから Ubuntu が起動する。

但し、このままでは端末を閉じると VM が強制終了してしまう。

これを避けるには tmux を使ってバックグラウンドで走らせる必要がある。

tmux を使うには、以下のようにスクリプトを編集すればよい。

```
# vi ubuntu0
```

CONSOLE= " default " を " tmux " または " tmux-detached " に

(tmux-detached は実行時に tmux を detach する設定)

Debian 7.3 の場合

VMOSVER を "debian7.3.0" にすれば、Ubuntu 13.10 とほぼ同様の手順でインストール可能です。

CentOS 6.5 の場合

```
# cd bhyve-script
```

```
# cp vm0 centos1
```

これから作る VM の名前に vm0 をコピー（末尾は数値で、他の VM と重複しない値でなければならない）

```
# vi centos1
```

NIC= " em0 " をお使いの NIC 名に
VCPUS= " 1 " を任意の vCPU 数に
VMRAM=1024 " を任意のメモリサイズに
VMOS= " freebsd " を " linux " に
VMOSVER= " 9.2-RELEASE " を " centos6.5 " に
DEVSIZE= " 2G " を任意のディスクサイズに
それぞれ変更。

```
# sh centos1 iso
```

ISO ファイルがダウンロードされ、ブートローダが起動する。

" Error processing drive: pci-0000:00:02.0-virtio-pci-virtio0 " というようなエラーが表示されたら「Re-initialize all」を選択する。

パーティション設定の画面では「Use entire disk」を選択する。

```
# sh centos1 start
```

HD イメージから CentOS が起動する。

FreeBSD 9.2-RELEASE の場合

```
# cd bhyve-script
```

```
# cp vm0 freebsd2
```

これから作る VM の名前に vm0 をコピー（末尾は数値で、他の VM と重複しない値でなければならない）

```
# vi freebsd2
```

NIC= " em0 " をお使いの NIC 名に

VCPUS= " 1 " を任意の vCPU 数に

VMRAM=1024 " を任意のメモリサイズに

DEVSIZE= " 2G " を任意のディスクサイズに

それぞれ変更。

```
# sh freebsd2 iso
```

ISO ファイルがダウンロードされ、ブートローダが起動する。

```
# sh freebsd2 start
```

HD イメージから FreeBSD が起動する。

OpenBSD 5.4 の場合

OpenBSD サポートはまだ実験段階で、カーネルを改変したバージョンの 5.4 しか動きません。そのため、以下のような手順で改変済みディスクイメージを取得・インストールする必要があります。

```
# cd bhyve-script
```

```
# cp vm0 openbsd3
```


まとめ

今回は FreeBSD 10.0-RELEASE で実際にいくつものゲスト OS を実行する方法を解説しました。いよいよ bhyve が実用的に使えるようになってきたのを実感して頂けたことと思います。次回は、今号で紹介する予定でした仮想マシンのネットワークデバイスについて解説します。

ライセンス

Copyright (c) 2014 Takuya ASADA. 全ての原稿データはクリエイティブ・コモンズ 表示 - 継承 4.0 国際ライセンスの下に提供されています。