bhyve zones in SmartOS

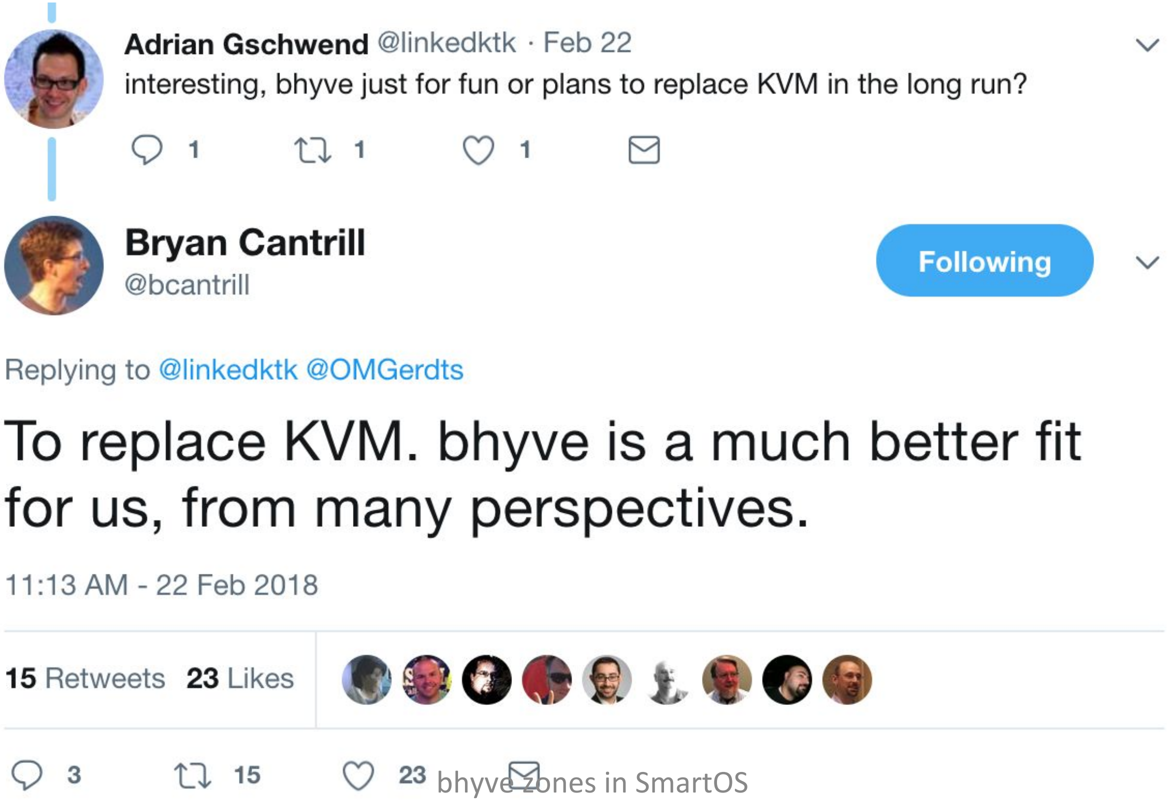
March 2018 development status
Written by Mike Gerdts
Presentation by Sam Gwydir

The Joyent bhyve-on-SmartOS Team

- Patrick Mooney
  - vmm, other kernel stuff, viona
- Hans Rosenfeld
  - Initial user space work, PCI passthrough
- Mike Gerdts
  - Zones integration
- Jerry Jelinek
  - Lots of fixes all over the place
- John Levon
  - Yet more fixes

- *Them, and…*
  - *Pluribus, who did the initial port*
  - *Lots of others that have helped debug and pound on development bits*

Why bhyve?



Adrian Gschwend @linkedktk · Feb 22

interesting, bhyve just for fun or plans to replace KVM in the long run?

Bryan Cantrill
@bcantrill

Replying to @linkedktk @OMGerdts

To replace KVM. bhyve is a much better fit for us, from many perspectives.

11:13 AM - 22 Feb 2018

15 Retweets **23** Likes

Timeline

- September 2018
  - Obtained Pluribus’ bhyve/illumos port
- February 2018
  - Passthru Support (GPU and NICs tested)
  - zhyve zone
  - vmadm integration

Difficulties

- The current vmm depends on a bunch of FreeBSD abstractions and thus requires a glue layer
- The core of bhyve is polished, but outer layers tend to make more assumptions

Added Features

- An API for registering external drivers
  - e.g. viona (network driver) can register a callback for notifications so traps outside kernel aren’t required, cutting down on syscalls.

Wants

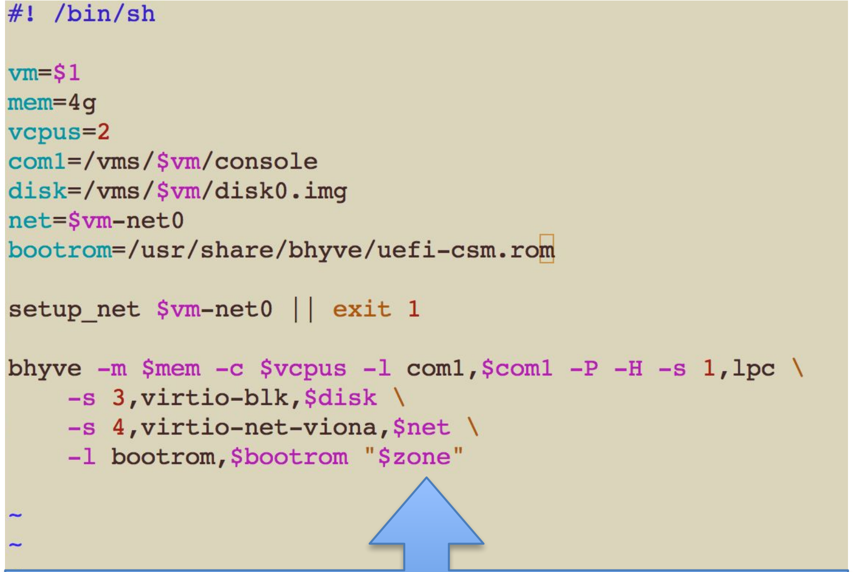
- Dynamic allocation for vcpu-related structures
  - Need support for 64 and 128 VCPUs
- Testing

Differences

- SmartOS does not have nmdm, we need more flexibility with use of UNIX domain sockets to connect to LPC COM devices
- UEFI EDK2 extended write support
  - SmartOS/bhyve only uses UEFI-CSM

Why in a zone?

- Convenient way to configure virtual resources, resource controls, and reduced privileges
- Defense in depth
  - Anti-spoofing
  - Escape bhyve into reduced privilege container with small attack surface
- Integrated with core OS features
- Higher-level tools already work well with zones



```
#! /bin/sh

vm=$1
mem=4g
vcpus=2
com1=/vms/$vm/console
disk=/vms/$vm/disk0.img
net=$vm-net0
bootrom=/usr/share/bhyve/uefi-csm.rom

setup_net $vm-net0 || exit 1

bhyve -m $mem -c $vcpus -l com1,$com1 -P -H -s 1,lpc \
    -s 3,virtio-blk,$disk \
    -s 4,virtio-net-viona,$net \
    -l bootrom,$bootrom "$zone"

~
~
```

Very basic script to start bhyve, but no integration with boot, resource controls, security isolation, etc.

bhyve zone highlights

File Systems

- / contains mountpoints, logs
- `/lib` and `/usr` read-only mounted from global zone
- `/dev`, with much removed
- A couple `tmpfs` file systems

Guest storage

- Configured with `device` resources
- Virtio driver
  - Others work, but not wired into zones configuration yet
- ZFS volumes

bhyve zone highlights

Networking

- Configured via `net` resources
- Automatic creation of vnics at boot, teardown on halt
- Anti-spoofing built-in
- Guest configuration via cloud-init

LPC devices

- Configurable bootrom, defaults to uefi-csm
- COM1
  - connected to `/dev/zconsole`
  - With proper console redirection, get to guest console with either of
    - `zlogin -C <zonename>`
    - `vmadm console <zonename>`

Zone boot & halt

boot

- Set up zone kernel context
  - `zone_t`, etc.
- Configure vnics
- Generate bhyve args
- Run zone `init` process, zhyve
  - Allocate & initialize resources
  - Signal that virtual HW setup complete
  - Run guest code

halt

- Destroy vmm instance
  - Free guest RAM
  - vmm has hold on `zone_t`
- Tear down virtual networking
- Tear down remaining zone context

Upcoming work

- Finish integrating our initial work into illumos-joyent master branch
- Get PCI passthrough hooked into bhyve brand
- Upstream bhyve and bhyve brand to illumos
  - Fair amount of prep work for this
- Resync with FreeBSD (and upstreaming)

Zones work FreeBSD may like

- State change notifications
  - So `zoneadmd` knows when virtual hardware allocation is successful
  - Better differentiation of guest halt vs. bhyve crash
- SMBIOS hacking
  - set system type, serial number, etc.
- UNIX domain sockets for serial ports & VNC
- mevent unit tests
- UEFI int13 extended write (LBA vs. C/H/S) support
- And that’s surely not all!

Updates

- Follow
  - Patrick Mooney: @pfmooney
  - John Levon: @johnlevon
  - Mike Gerdts: @OMGerdts
- Blog https://mgerdts.github.io/
  - Atom feed: https://mgerdts.github.io/feed.xml
- Github: https://github.com/joyent/illumos-joyent