OpenBSD vmm/vmd Update

Mike Larkin

bhyvecon 2016
11 Mar 2016 – Tokyo, Japan

Agenda

- History and overview of vmm / vmd
- Current status
- Future plans
- (Reyk): Improvements to vmd / vmctl
- Q&A

VMM History

- People have wanted a native OpenBSD hypervisor for some time
- One night, someone bought me a beer and challenged me to build one ...

VMM History

- People have wanted a native OpenBSD hypervisor for some time
- One night, someone bought me a beer and challenged me to build one ...
  - Isn't this how all these stories start?

VMM History

- Started coding at Brisbane 2015 hackathon
- Solo development through the summer and fall
  - Thank OpenBSD Foundation for a grant to support this work
- First commits late fall 2015

VMM History

- Why not just port bhyve?
- I Looked at this …
- Equal effort to port or rewrite
  - Seemed to be different project goals anyway
  - We wanted legacy support, i386, etc...

VMM Initial Design Goals

- “Make it work, make it right, make it fast”
- Support different processor models
  - Support advanced processor features, but don't require them
  - Support i386
- Get OpenBSD on OpenBSD working first
  - Then “generic virtio based VM”
  - Work on other things later

VMM Overview

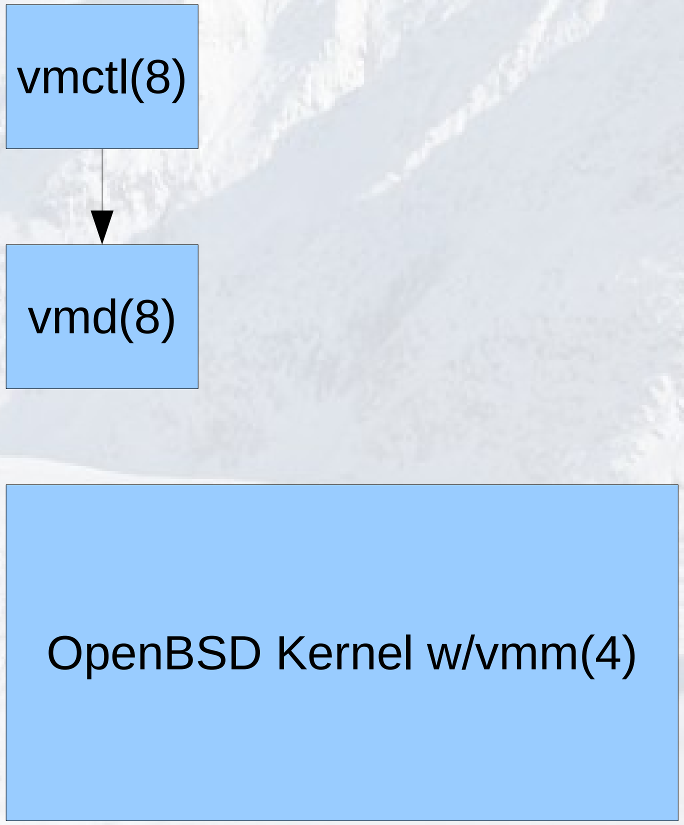
- VMM has several parts
- vmd(8)
  - User mode daemon
  - Makes requests to vmm(4) to run VMs
  - Handles virtual device I/O

VMM Overview

- vmm(4)
  - In-kernel part
  - Executes guest VM code
  - Transfers control to vmd(8) when device I/O or interrupts occur
- vmctl(8)
  - User mode control program
  - Starts, stops, and controls VMs

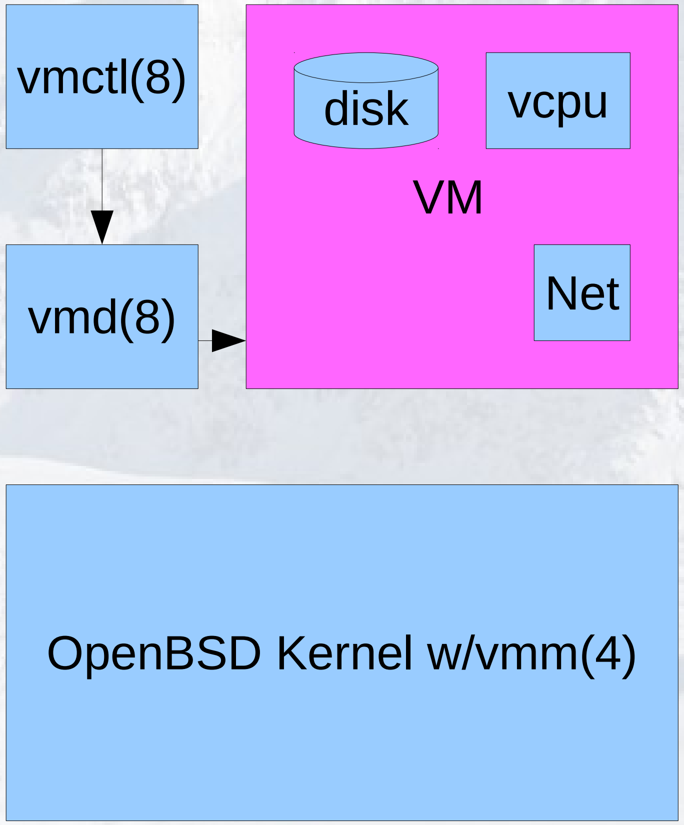
VM Execution

- A user creates a VM
  - “vmctl start ...”



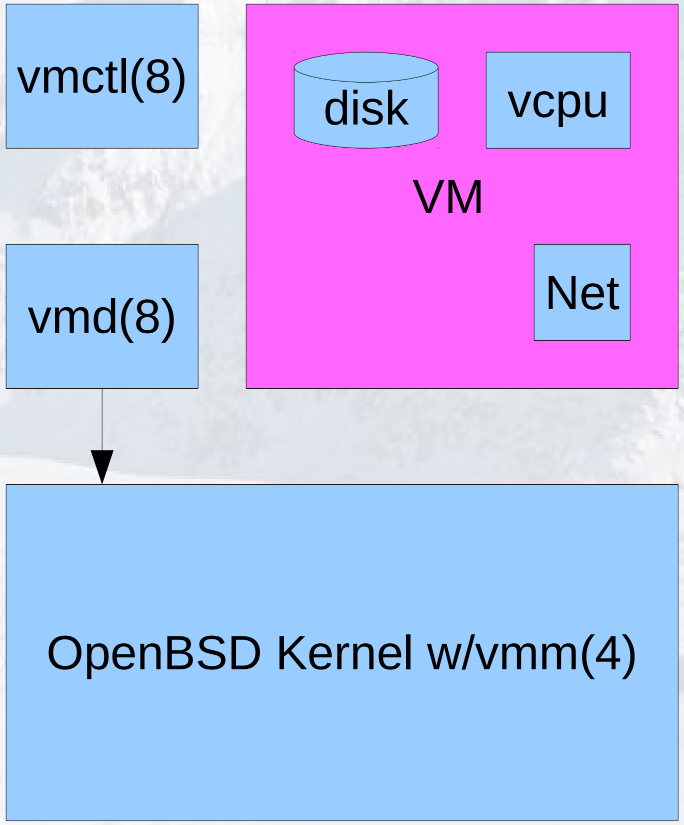
VM Execution

- A user creates a VM
    - “vmctl start ...”
- vmctl asks vmd to create VM with requested devices



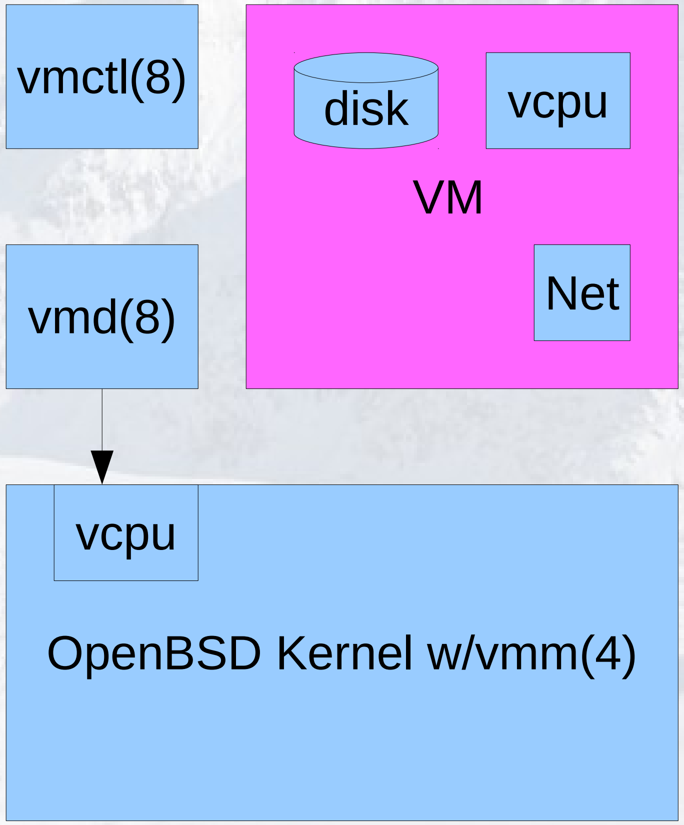
VM Execution

- vmd asks vmm to run the VM (for each vcpu)



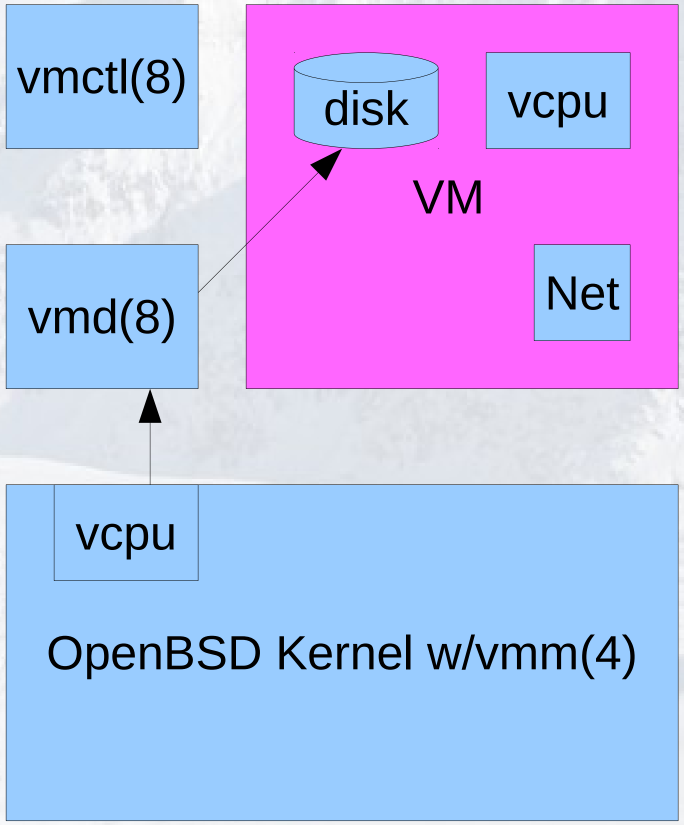
VM Execution

- vmd asks vmm to run the VM (for each vcpu)
- vmm runs the vcpu until help required (exit)
  - Device I/O
  - Memory allocation
  - Interrupt
  - Etc...



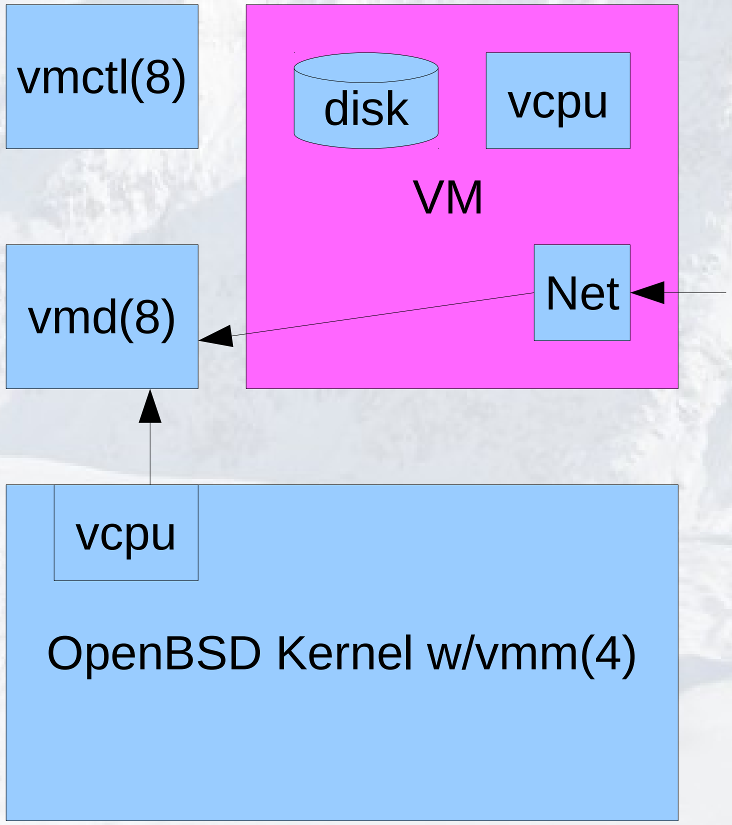
VM Execution

- Control returns to vmd as needed
  - Device I/O (Disk)



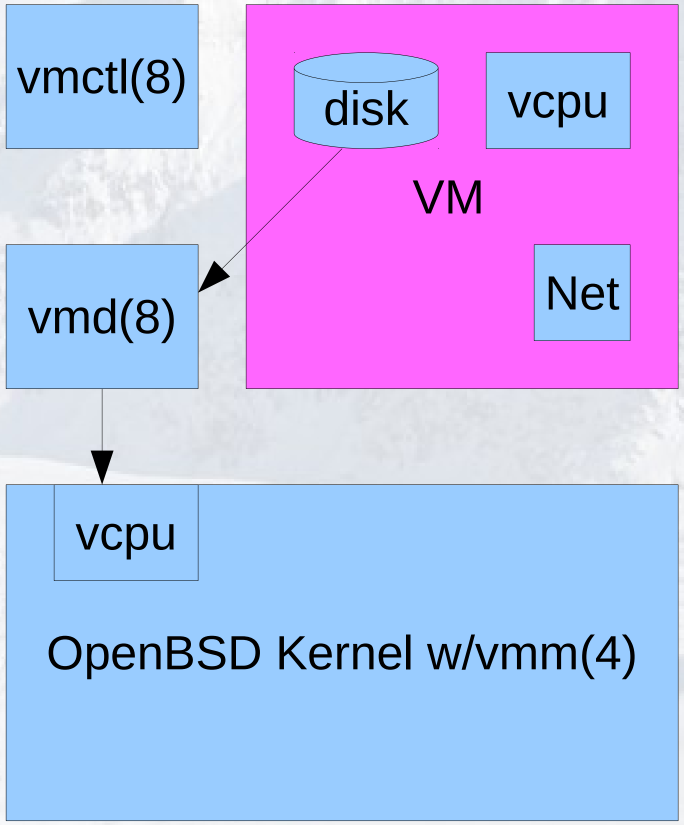
VM Execution

- Control returns to vmd as needed
    - Device I/O (Network)



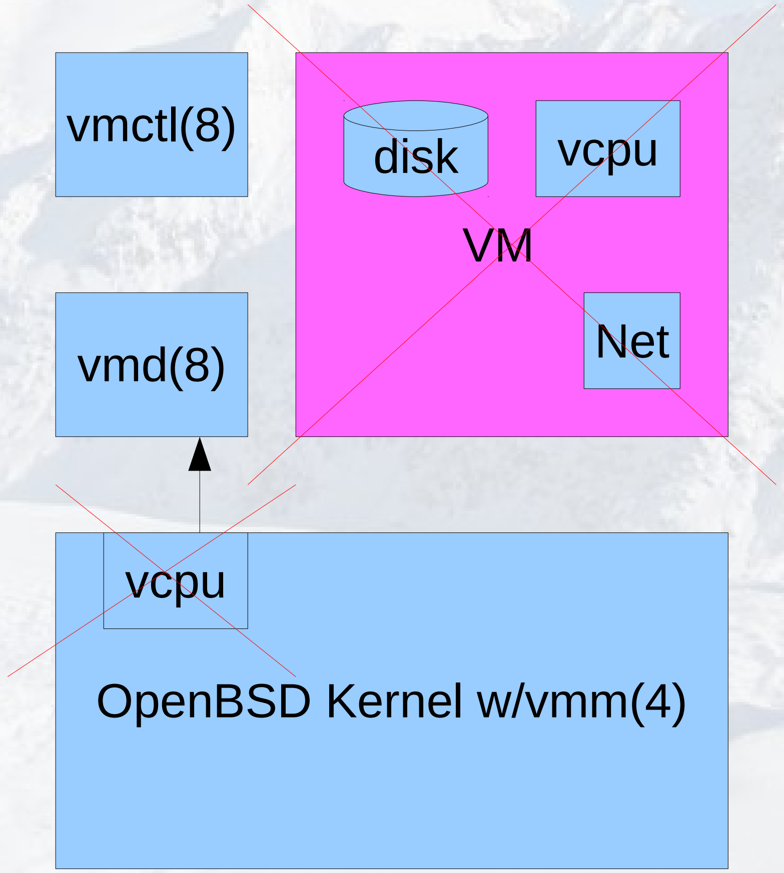
VM Execution

- Control returns to vmd as needed
  - Device I/O
- vmd performs the I/O operation
  - Repeat vcpu launch ...



VM Execution

- Control returns to vmd as needed
  - Prohibited operations
  - VM termination



Current Status

- Device model
  - Serial conosle
  - virtio(4) devices
    - vio(4) for networking
    - vioblk(4) for disks
  - Platform devices (legacy devices) as needed

Current Status

- VM compatibility
  - Initial focus on amd64 OpenBSD guests
  - vmd(8)'s boot loader can load arbitrary ELF kernels
  - I loaded both FreeBSD and NetBSD (not currently a personal priority)

Current Status

- After initial commit, many other developers became involved
  - Some working on vmm
  - Some working on vmd/vmctl
- My initial vmd/vmctl code sucked

Current Status

- After initial commit, many other developers became involved
  - Some working on vmm
  - Some working on vmd/vmctl
- My initial vmd/vmctl code sucked
  - So Reyk stepped in to fix things

Current Status

- After initial commit, many other developers became involved
  - Some working on vmm
  - Some working on vmd/vmctl
- My initial vmd/vmctl code sucked
  - So Reyk stepped in to fix things
  - I probably owe him a beer

Future Plans

- vmm(4) features
  - Nested VMX
  - i386
  - AMD SVM
    - Then someone will ask for nested SVM …
- All these are implemented to some degree, rotting in my tree

Future Plans

- VM templates
  - vmctl run firefox
    - Boots firefox in a VM
    - Filesystem passthrough with whitelist
      - Eg, to let firefox access host ~/.mozilla
    - Forwarded display, isolated network
- VM migration
  - vmctl send “myvm” | ssh mlarkin@foo.com vmctl receive

Future Plans

- One developer is working on qemu interface
  - For legacy OS support
- One developer is working on making vmm look like KVM
  - Easier interfacing with existing tools (also gives another route to qemu interface)

Finally ...

- If you want to get involved…
  - ... find something interesting (or ask what needs to be done)
  - … implement it
  - … send a diff

Questions?

- Any questions?

Thank You

Mike Larkin
mlarkin@openbsd.org
@mlarkin2012