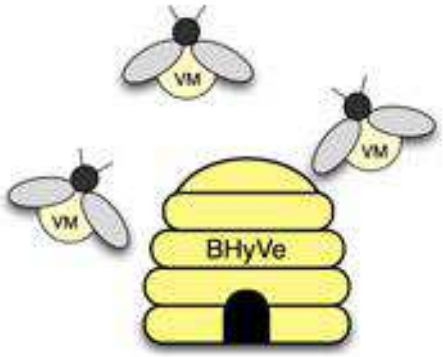


Paul Schenkeveld

"The bhyve hypervisor"

FOSDEM

February 3, 2013

About the speaker

Where am I and what am I doing in this basket?

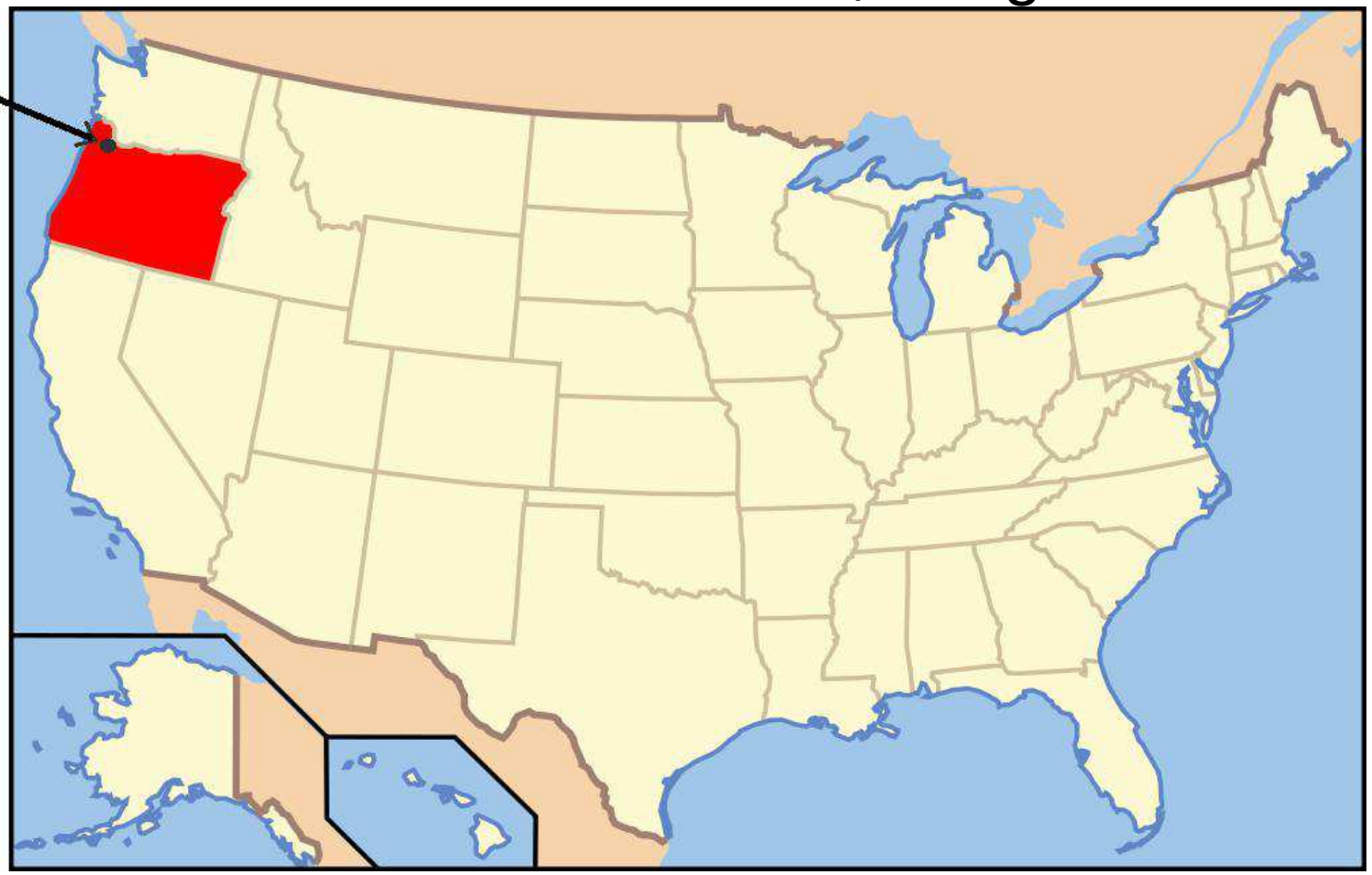
Michael Dexter

Michael published about ***bhyve*** on callfortesting.org and gave many talks about ***bhyve***.
He was invited to talk here, but ...

Michael Dexter

but ...

Michael lives in Portland, Oregon



Fosdem 2013 - bhyve

Paul Schenkeveld

So you'll have to listen to me today

I've been working with FreeBSD from almost the beginning, but am relatively new to ***bhyve***.

As a long time friend of Michael I've been helping him working with ***bhyve*** last year but didn't have the hardware to run ***bhyve*** myself.

I dove into ***bhyve*** last month when Michael asked me to stand in for him.

Hypervisors

Robert P. Goldberg, 1973:

"Architectural Principles for Virtual Computer Systems"

- Type 1 Native/Bare Metal Hypervisor (VMware)
- Type 2 Hosted Hypervisor (Xen, KVM)

Gerald J. Popek and Robert P. Goldberg, 1974:

"Formal Requirements for Virtualizable Third Generation Architectures"

Properties of a Hypervisor:

http://en.wikipedia.org/wiki/Hypervisor

http://en.wikipedia.org/wiki/\
Popek_and_Goldberg_virtualization_requirements

Context

We already have many hypervisors:

- Proprietary and dominant: VMware (GPL Violations?)
- Linux: KVM/QEMU Hypervisor / LXC Containers
- SmartOS: KVM Hypervisor / Zones
- FreeBSD: Xen EC2 / jail(8)
- Honorable mention: NetBSD Xen
- Virtualbox

Do we need another Hypervisor?

MeetBSD California 2010

- 100% Virtualization session attendance
- Follow-up session the next day

Conclusion:

- We need a BSD hypervisor!

Context

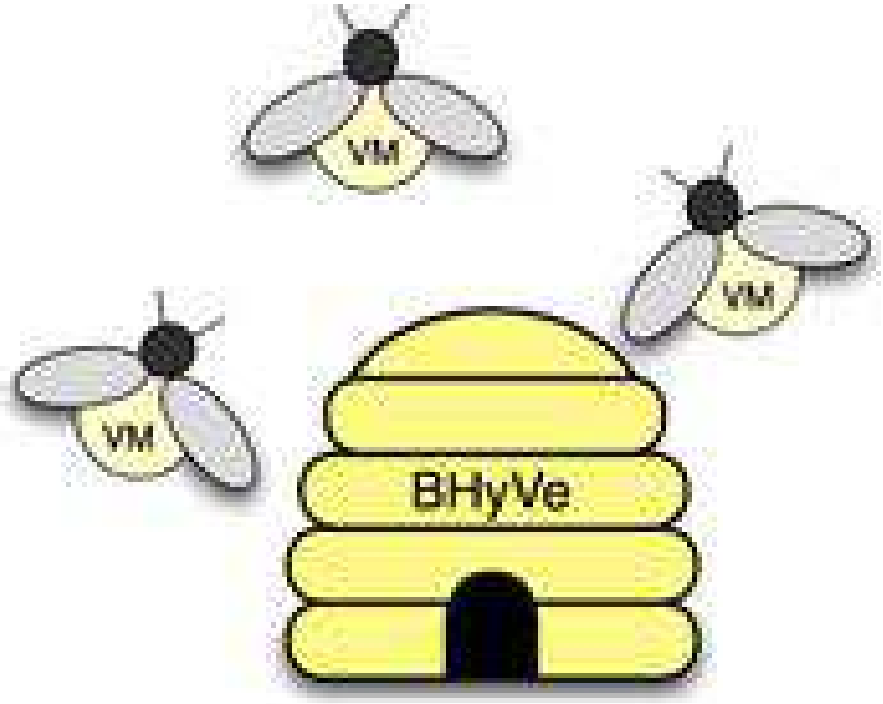
- Equivalence / Fidelity
  A program running under the VMM should exhibit a behavior essentially identical to that demonstrated when running on an equivalent machine directly.
- Resource control / Safety
  The VMM must be in complete control of the virtualized resources.
- Efficiency / Performance
  A statistically dominant fraction of machine instructions must be executed without VMM intervention.

http://en.wikipedia.org/wiki/\
Popek_and_Goldberg_virtualization_requirements

bhyve - The "BSD HyperVisor"

Written by:

- Neel Natu
- Peter Grehan



First announcement:

- BSDcan 2011 Devsummit

Slides and audio are online:

- http://wiki.freebsd.org/201105DevSummit?action=AttachFile&do=view&target=BHyVe.pdf
- http://www.bsdcan.org/2011/audio/\BHyVeNativeBSDHypervisor.mp3

bhyve - The "BSD HyperVisor"

- Requires Intel Extended Page Tables (EPT)
- Guests are booted from disk images
- jhb@'s `as(1)` fixes allow for a GPLv2 assembler with EPT
- Easiest to test on 9.0 and 10-CURRENT (Not 8.x or 9.1)
- 8.3 guests feasible, 9.0 and newer supported
- Minor fixes planned to allow for building with CLANG (***done***)
- Works with VMware Fusion VT-x pass-through (on Mac)

- Imported to HEAD (aka. -CURRENT, FB-10) last month!

Hardware-Assisted Virtualization

"It’s all built on VT-x exits. I/O exits are used to build the PCI emulation since I/O instructions are used for PCI config space. VT-x sets up state to enter/exit "non-root" mode, aka VM mode.

EPT-violation exits are used for memory-mapped I/O. Guest physical memory is set up in EPT tables. Guest access to anything else causes an EPT-violation exit. Then instruction emulation is used to determine what is written and where reads should go."

– Peter Grehan

Hardware-Assisted Virtualization

- VT-x: Virtualization Extensions:
  Interception of privileged instructions
- VT-d: Virtualization for Directed I/O:
  IOMMU Virtualization / PCI Pass-Through
- EPT: Extended Page Tables: MMU Virtualization
  Previously handled in software
- AMD-V, AMD-Vi and RVI/Nested Page Tables Planned

http://en.wikipedia.org/wiki/X86_virtualization
http://en.wikipedia.org/wiki/Extended_Page_Table

Does my hardware support bhyve?

Look at the dmesg(8) output:

Features=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,
MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,
CLFLUSH,DTS,ACPI,MMX,FXSR,SSE,SSE2,SS,HTT,TM,PBE>
Features2=0x17bae3ff<SSE3,PCLMULQDQ,DTES64,MON,
DS_CPL,VMX,SMX,EST,TM2,SSSE3,CX16,xTPR,PDCM,
PCID,SSE4.1,SSE4.2,x2APIC,POPCNT,TSCDLT,AESNI,
XSAVE,AVX>

- POPCNT Accompanies EPT

Hardware-Assisted Virtualization

Most, if not all "Nehalem", "Sandy Bridge" and "Ivy Bridge" Core, Xeon, Pentium and Celeron Processors:

- EPT: Intel Core i3, i5, i7 Processors
- EPT: Most same-generation Xeon Processors
- EPT: Some Pentium Mobile and Celeron Processors (!)
- VT-d: Many Core i5, i7 and Xeon Processors

Caveat: It may be disabled in BIOS or blocked entirely

Hardware-Assisted Virtualization

ark.intel.com is your friend:

Intel® Pentium® 2020M Processor (2M Cache, 2.40 GHz)

Advanced Technologies	
Intel® Turbo Boost Technology	No
Intel® vPro Technology	No
Intel® Hyper-Threading Technology	No
Intel® Virtualization Technology (VT-x)	Yes
Intel® Virtualization Technology for Directed I/O (VT-d)	No
Intel® VT-x with Extended Page Tables (EPT)	Yes

as(1) Implications

(only on FreeBSD-9)

- The GNU binutils assembler with the EPT instructions is GPLv3-licensed and will not be included in BASE
- 9.0 Short-term solution: Install `devel/binutils` (v22.2_3)
- HEAD solution: jhb@ implemented the missing instructions and they are manually MFC-able

  See FreeBSD svn revisions 238123 and 238167

as(1) Implications

Revision 238123

Add support for the ’xsave’, ’xrstor’, ’xsaveopt’, ’xgetbv’, and ’xsetbv’ instructions. I reimplemented this from scratch based on the Intel manuals and the existing support for handling the fxsave and fxrstor instructions. This will let us use these instructions natively with GCC rather than hardcoding the opcodes in hex.

Revision 238167

Add support for the ’invept’ and ’invvpid’ instructions. Beyond simply adding appropriate table entries, the assembler had to be adjusted as these are the first non-SSE instructions to use a 3-byte opcode (and a mandatory prefix to boot).

The Versioning Moving Target

- ***bhyve*** began life in FreeBSD 8.1
- Is manageable in FreeBSD 9.0
- FreeBSD 9.1 suffers from AVX floating point changes
- FreeBSD 10 projects/bhyve is^W was the official home
- ***bhyve*** is included in FreeBSD-10 since January 2013

My recommendation:

- Use a recent FreeBSD-10
- Just works out-of-the-box

bhyve Host Components

- `/usr/sbin/bhyve`
  the user-space sequencer and I/O emulation
- `/usr/sbin/bhyveload`
  the user-space FreeBSD loader
- `/usr/sbin/bhyvectl`
  a utility to dump hypervisor register state
- `/usr/lib/libvmmapi.{a,so*}`
  the front-end to the vmm.ko chardev interface
- `/boot/kernel/vmm.ko`
  kernel module for VT-x, VT-d and hypervisor control
- `/boot/kernel/if_tap.ko`
  not part of ***bhyve*** but needed for networking

bhyve Guest Kernel Components

Your guest system needs the following kernel modules to interact with the hypervisor:

- device virtio # Generic VirtIO bus (required)
- device virtio_pci # VirtIO PCI device
- device vtnet # VirtIO Ethernet device
- device virtio_blk # VirtIO Block device
- device virtio_scsi # VirtIO SCSI device
- device virtio_balloon # VirtIO Memory Balloon device

These are included in GENERIC since ***bhyve*** import but can also be loaded as modules.

Host Preparation

Add to `/boot/loader.conf`:

```
# Deduct memory from the host for the guests
# (only on FreeBSD-9), reboot to take effect
hw.physmem="0x100000000"        # 4GB for the host
#hw.physmem="0x200000000"       # 8GB for the host

# Suppress noise (or build non-debugging kernel):
debug.witness.watch="0"

# Load modules (loadable with kldload too):
vmm_load="YES"
if_tap_load="YES"
bridgestp_load="YES"
if_bridge_load="YES"
```

DIY bhyve

http://people.freebsd.org/~neel/bhyve/bhyve_instructions.txt

- Make sure that you are running a recent version of FreeBSD 10.0 that includes ***bhyve*** support (i.e. anything after r245652).
- Load the following kernel modules:

  kldload vmm

  kldload if_tap
- Create the tap0 interface

  ifconfig tap0 create

DIY bhyve

- Download 'vmrun.sh' and 'release.iso' from
  http://people.freebsd.org/~neel/bhyve

  Copy both into the same directory in which you have write permission.

  Please be aware that my 'release.iso'' is most likely stale and you can get the latest release.iso here:

  ftp://ftp.freebsd.org/pub/FreeBSD/snapshots/\
  amd64/amd64/ISO-IMAGES/10.0/

- As root (or using sudo) execute the following commands:

```
./vmrun.sh vm1
```

DIY bhyve

- Select default console type "`vt100`" or whatever is appropriate for you. Install on disk device `vtbd0` (appears as a 8GB disk device). At the end select "yes" when the "Manual Configuration" box appears.

Type in the following:

```
cat >> /etc/ttys << EOF
console "/usr/libexec/getty std.9600"   vt100   on secure
EOF
```

And then reboot

- Enjoy your virtual machine

DIY bhyve

And then you see:

```
Launching virtual machine "vm1" ...
Consoles: userboot

FreeBSD/amd64 User boot, Revision 1.1
(paul@bhyve.psconsult.nl, Sun Jan 27 17:53:58 CET 2013)
Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0xd00880 data=0x15ed20+0x2bbc90 syms=[0x8+0x14c2c8+0x8+0x1a8066]
/boot/kernel/virtio.ko size 0x5a50 at 0x1810000
/boot/kernel/virtio_pci.ko size 0x6d68 at 0x1816000
/boot/kernel/virtio_blk.ko size 0x6a30 at 0x181d000
/boot/kernel/if_vtnet.ko size 0xd8e8 at 0x1824000
\
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [/boot/kernel/kernel]...
GDB: no debug ports present
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2012 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 10.0-CURRENT #23 r243595:243640: Tue Nov 27 17:11:16 PST 2012
    neelnatu@neelnatu4:/usr/obj/usr/freebsd/projects/bhyve/sys/GENERIC amd64
WARNING: WITNESS option enabled, expect reduced performance.
CPU: Intel(R) Xeon(R) CPU           E5620  @ 2.40GHz (2399.90-MHz K8-class CPU)
  Origin = "GenuineIntel"  Id = 0x206c2  Family = 0x6  Model = 0x2c  Stepping = 2
  Features=0x8fa3ab7f<FPU,VME,DE,PSE,TSC,MSR,PAE,CX8,APIC,SEP,PGE,CMOV,PAT,PSE36,DTS,MMX,FXSR,SSE,SSE2,SS,PBE>
  Features2=0x80bee255<SSE3,DTES64,DS_CPL,SMX,SSSE3,CX16,xTPR,PDCM,PCID,DCA,SSE4.1,SSE4.2,x2APIC,POPCNT,HV>
  AMD Features=0x2c100800<SYSCALL,NX,Page1GB,RDTSCP,LM>
  AMD Features2=0x1<LAHF>
  TSC: P-state invariant
real memory  = 2147483648 (2048 MB)
avail memory = 2042015744 (1947 MB)
```

DIY bhyve

To stop your running guest:

- Type "`reboot`" or "`shutdown -r`" inside the guest
- Press "ESC" at the loader and type "`quit`"

Legitimate Concerns

"x86 virtualization is about basically placing another nearly full kernel, full of new bugs, on top of a nasty x86 architecture which barely has correct page protection. Then running your operating system on the other side of this brand new pile of sh***t.***

You are absolutely deluded, if not stupid, if you think that a worldwide collection of software engineers who can't write operating systems or applications without security holes, can then turn around and suddenly write virtualization layers without security holes."

– Theo de Raadt

Legitimate Concerns

"But ***bhyve*** code (kernel + userland libraries and utilities) adds only 250KB of source to the existing FreeBSD code base."

– Me

The ***bhyve*** heavy lifting is done by:
`/usr/src/sys/amd64/vmm/intel/vmx.c`
(1,300 LOC)

– Michael

The Future

What's coming:

- ACPI Tables
- AHCI Device Emulation
- VirtIO MSIx Support

On the horizon:

- Takuya Asada's BIOS Emulation Work
  Allows to run other operating systems as guest.
- AMD Support - In progress

The Future

On the horizon:

- Better Integration with Host Scheduler
- Memory Over-Commit
- Suspend and Resume
- Generalization of CPUID Features for Guest Portability
- Sparse Images (QCOW, VDI, VMDK, ZVOL?)
- Non-tap VirtIO Back-End

More info:

The main places for ***bhyve*** information are:

- http://bhyve.org/
- https://wiki.freebsd.org/BHyVe
- http://callfortesting.org/bhyve/
  Here you can also find a package to install ***bhyve*** on FreeBSD-9 easily
- The FreeBSD virtualisation mailing list, subscribe at
  http://lists.freebsd.org/mailman/listinfo/freebsd-virtualization

Manual pages are not complete (yet)

- Any volunteers to help writing docs?

Acknowledgements

Thank you:

* Neel and Peter for this beautiful piece of code!
* Michael for his endless testing, publications and presentations and for asking me to stand in
* The FOSDEM organisation for inviting this talk
* Ken Thompson, Dennis Ritchie et. al. for giving us UNIX and C.
* The CSRG team at Berkeley University for BSD.
* The FreeBSD project for FreeBSD.
* You, for not making me look like a fool talking to an empty room.
* All others who I should thank for making this talk possible but who are not yet mentioned above.

Questions?

You should have asked: "How can I help?"

Final words

Thank you and "Happy hacking"!