Updating bhyve UEFI Firmware & UEFI HTTP Boot
Agenda

- What is UEFI and what does it have to do with bhyve?
- Why does it need updated?
- How did I update it?
- How can we keep it up to date in the future?
- UEFI HTTP Boot
What is UEFI?

“This Unified Extensible Firmware Interface … Specification describes an interface between the operating system and the platform firmware.”

UEFI Specification, Version 2.8, §1

Figure 1. UEFI Conceptual Overview
What is EDK II?

TianoCore: community that supports EDK II

EDK II: (EFI Dev Kit 2) cross platform development environment for UEFI.
SPDX-License-Identifier: BSD-2-Clause.

OVMF: Open Virtual Machine Firmware, for QEMU and Xen. One of the several platforms supported by EDK II.
What is uefi-edk2-bhyve?

Adaptation of OVMF for bhyve.

uefi-edk2-bhyve = OVMF - QEMU + bhyve.

© 2014,2015

(btw, Not the only way to run a VM in bhyve).
Why do we need to update it?

UDK2014.SP1 ← UEFI 2.4B specification, April 2014
edk2-stable201903 ← UEFI 2.7 specification, May 2017

- new ACPI/SMBIOS versions
- Bug fixes/ performance improvements
- HTTP Boot
How did I update it?

1. Archaeology, find original branch point.

from

commit 28b621aeb5986b648182448c9c31e4e16e5528a9
Author: Peter Grehan <grehan@nahannisys.com>
Date: Fri Oct 16 22:44:53 2015 -0700

Initial import of bhyve UEFI patch.

125 files changed, 31426 insertions(+), 3 deletions(-)

to

commit c46543b586a14e288245ef6ad31b4e85152d6839
Author: Peter Grehan <grehan@nahannisys.com>
Date: Fri Oct 16 22:44:53 2015 -0700

Initial import of bhyve UEFI patch.

[d.scott.phillips@intel.com] This is an attempt to reconstruct the history of the bhyve patch by rephrasing it as a delta against OvmfPkg and unsquashing various upstream commits that were squashed into it. In particular, this is commit:

28b621aeb5 ("Initial import of bhyve UEFI patch.")

with the following commits unsquashed from it:

75 files changed, 17424 insertions(+), 1210 deletions(-)
How did I update it?

2. Move forward minimized changeset one release at a time
   → UDK2014.SP1 → UDK2015 → UDK2017 → UDK2018
   → edk2-stable201808 → edk2-stable201811
   → edk2-stable201903

Making sure boot/basic functionality keeps working each step of the way.
How did I update it?

Intermediate steps preserved:


- wip/2019-03/bhyve-history-reconstruct (doesn’t work)
- wip/2019-03/bhyve-rebase-UDK2014.SP1
- wip/2019-03/bhyve-rebase-UDK2015
- wip/2019-03/bhyve-rebase-UDK2017
- wip/2019-03/bhyve-rebase-UDK2018
- wip/2019-03/bhyve-rebase-edk2-stable201808
- wip/2019-03/bhyve-rebase-edk2-stable201811
- wip/2019-03/v2-bhyve-rebase-edk2-stable201903

(may help debugging at some point)
How will we keep it up to date in the future?


- Update OpenSSL version from 1.1.0j to 1.1.1b
- Add new toolchain for LLVM/CLANG8.0
- Replace BSD 2-Clause License with BSD + Patent Licence

More frequent, smaller updates should be less work overall.

Upstreaming? License?

How/who/when? thoughts?
Call to action: Help test it!

Help find & squish bugs!

ports: sysutils/uefi-edk2-bhyve-devel

thanks araujo@ & bcran@
UEFI HTTP Boot
**UEFI PXE Boot**

network boot standard adapted from the BIOS days

DHCP server tells you where TFTP server is

TFTP server gives you something to run

Either a UEFI binary, or a UEFI-compliant file system image

---

**Figure 2-1 PXE Boot**
UEFI HTTP Boot

Replace TFTP with HTTP

DHCP could return information to direct HTTP Boot...or not.

DNS could be used...or not.
EDK II’s RamDiskDxe publishes ramdisks to the OS using ACPI 6.0’s NFIT.
nvdimm.ko can consume the NFIT and present the ramdisk as a device to the system.
UEFI HTTP Boot - Demo
UEFI Interactive Shell v2.2
EDK II
UEFI 02.70 (BHYVE, 0x00010000)
map: No mapping found.
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> exit
Select Language <Standard English>

- Device Manager
- Boot Manager
- Boot Maintenance Manager

Continue
Reset

↑↓=Move Highlight  <Enter>=Select Entry

This selection will take you to the Device Manager
Network Device List

MAC: 00: A0: 98: BD: 63: 32

Press ESC to exit.
Network Device

- VLAN Configuration
- IPv4 Network Configuration
- **HTTP Boot Configuration**

Press ESC to exit.

Configure HTTP Boot parameters.
(MAC: 00:0A:98:BD:63:32)
HTTP Boot Configuration

<table>
<thead>
<tr>
<th>Input the description</th>
<th>FreeBSD</th>
<th>A new Boot Option will be created according to this Boot URI.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Protocol</td>
<td><code>&lt;IP4&gt;</code></td>
<td></td>
</tr>
<tr>
<td>Boot URI</td>
<td><a href="http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/amd64/ISO-IMAGES/12.0/FreeBSD-12.0-RELEASE-amd64-bootonly.iso">http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/amd64/ISO-IMAGES/12.0/FreeBSD-12.0-RELEASE-amd64-bootonly.iso</a></td>
<td></td>
</tr>
</tbody>
</table>

**Keyboard Shortcuts:**
- F9 = Reset to Defaults
- F10 = Save
- Esc = Exit
- ↑ = Move Highlight
- <Enter> = Select Entry

Configuration changed
Select Language  <Standard English>  This selection will take you to the Boot Maintenance Manager

- Device Manager
- Boot Manager
- **Boot Maintenance Manager**

Continue
Reset

↑↓=Move Highlight  <Enter>=Select Entry
Welcome to FreeBSD

1. Boot Multi user [Enter]
2. Boot Single user
3. Escape to loader prompt
4. Reboot

Options:
5. Kernel: default/2kernel (1 of 1)
6. Boot Options

Exiting menu!

Type '?' for a list of commands, 'help' for more detailed help.
OK set boot_serial=no
OK set boot_verbos=yes
OK load /boot/kernel/kernel
/boot/kernel/kernel text=0x1670a60 data=0x1cd380+0x760b40 syms=[0x0+0x174c46+0x0+0x1922a1]
OK load /boot/kernel/nvmm.ko
/boot/kernel/nvmm.ko size 0x7098 at 0x263d000
OK boot_