Intel HAXM – a hardware-assisted acceleration engine in the NetBSD kernel

AsiaBSDCon 2019
Tokyo, Japan

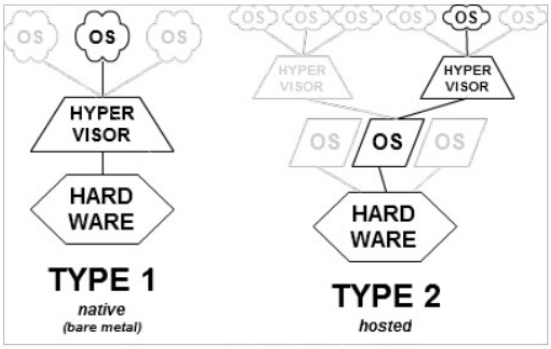
Kamil Rytarowski
The NetBSD Foundation
kamil@NetBSD.org

Abstract

Intel HAXM (Hardware Accelerated Execution Manager) is a hypervisor that works as a loadable kernel module in multiple kernel environments. HAXM uses the Intel Virtualization Technology (VTx) and thus requires an Intel hardware architecture. A hypervisor (on a so called host machine) is a piece of software or hardware (sometimes both) that can create and run a virtual machine (called a guest machine). Usage of virtual machines reduces the need for using each software or product on dedicated hardware and can fully isolate it from the environment, which makes it suitable for data migration, reduction of costs of running multiple instances of guest machines on a host machine.

1. Introduction

There are two types of hypervisors: Type 1 and Type 2. The first type requires booting directly into a hypervisor that is a standalone program and it can be utilized to spawn guest machines, while one of them typically is a master guest that can orchestrate the hypervisor. The second type allows to boot into the default kernel of the host and the hypervisor runs as a kernel module in kernel space, creating dedicated guest-machines in a privileged kernel-space on demand.



[source: wikipedia; license: CC0]

2. Xen

The NetBSD Operating System traditionally uses Xen virtualization (NetBSD/xen). Xen is a Virtual Machine Monitor that was integrated with the distribution sources in 2004. The Xen technology is supported by NetBSD/i386 (x86 32-bit) and NetBSD/amd64 (x86_64). Over the years the set of features and hardware facilities have evolved and are still maintained by the NetBSD developers, although there is no full feature parity with other kernels.

A shortcoming of Xen/NetBSD, besides the lacking parity in features is that it’s not a convenient solution for typical desktop usage. There are reported conflicts with the X Window system, which is nowadays a mandatory graphical system on UNIX systems.

Another property that makes the usage of XEN more difficult is the need to redefine the booting process and directly start a standalone hypervisor that manages all guests. This is a Type 1 hypervisor. On a desktop computer a user usually prefers by far to enable or disable features without the need to reboot and especially without the need to reinstall the Operating System.

3. Type 2 (hosted) virtualization

Nowadays desktop setups tend to use Type 2 virtualization more often which is less intrusive for the host system, however there wasn't any available solution so far that would be supported by NetBSD. A popular front-end to hypervisors of this type is Qemu – a GPLv2 software maintained by the Open-Source community. The primary purpose of Qemu is to utilize software-assisted emulation (softemu) or hardware-assisted emulation. The software-assisted emulation is required for emulating a different CPU than the underlying host CPU, however for the use cases of emulating the same type of architecture there is room for hardware-assisted emulation that can offer a massive boost in speed of code execution inside guest machines.

4. Qemu with hardware-acceleration

There is a number of hypervisor solutions for mainstream Operating Systems that work as plugins for qemu. The most important ones are:

- KVM for Linux,
- HVF for Darwin,
- WHPX for Windows.

Until the process of making the HAXM project available to the Open-Source community there wasn’t a single solution for the mentioned systems, basically due to the fact that each of these three systems weren’t from the same kernel family and each requires specific knowledge. This also resulted in the solution of each of the mentioned systems that were closely tied to each of the specific kernels and hardly reusable by someone else.

5. HAXM

Intel HAXM (Hardware Accelerated Execution Manager) is a hypervisor that works as a loadable kernel module in multiple kernel environments. HAXM uses the Intel Virtualization Technology (VTx) and thus requires an Intel hardware architecture.

Until the process of making the HAXM project available to the Open-Source community there wasn’t a single solution for the mentioned systems, basically due to the fact that each of these three systems weren’t from the same kernel family and each requires specific knowledge. This also resulted in the solution of each of the mentioned systems that were closely tied to each of the specific kernels and hardly reusable by someone else.

I decided to give a HAXM port to NetBSD a try as it was already available for two kernel families: Windows and Darwin, which is rather an unusual pair of supported kernels by an open-source project. The process of porting HAXM to NetBSD however was still difficult as semantics of internals of these systems were alien to NetBSD internals. This has suspended the porting process for a while, until the moment when Linux was supported as host, which exposed internals that are well documented in the OpenSource resources and much closer to the model of the NetBSD kernel, and thus more easily mappable into native code. A HAXM port to the NetBSD kernel has been ported, which involves the kernel workaround patch and qemu patching. The final deliverable is that the HAXM engine is successfully emulating NetBSD, Linux and Windows as guests, while support for other systems is either functional or close to be so.

With the advent of Intel's open-sourcing of their HAXM engine, we now have access to an important set of features:

- A BSD-style license.
- Support for multiple platforms: Windows, Darwin, Linux, and now NetBSD .
- HAXM is an Intel hardware assisted virtualization for their CPUs (VTx and EPT needed).
- Support for an arbitrary number of concurrent VMs. For simplicity's sake, NetBSD only supports 8, whereas Windows/Darwin/Linux support 64.
- An arbitrary number of supported VCPUS per VM. All OSs support up to 16 VCPUs.
- ioctl(2) based API (/dev/HAX, /dev/haxm_vm/vmXX, /dev/haxm_vmXX/vcpuYY).
- Implement non-intrusively as an out-of-tree, standalone executable kernel module.
- Default compatibility with qemu as a front-end.
- Active upstream support from Intel, which is driven by commercial needs.
- Optimized for desktop scenarios.
- Probably the only open-source cross-OS virtualization engine.
- An active and passionate community that is dedicated to improving it.

As well as a few of HAXM's downsides:

- No AMD (SVM) support (although there are community plans to implement it).
- No support for non-x86 architectures.
- Need for a relatively recent Intel CPU (EPT required).
- Not as flexible as KVM-like solutions for embedded use-cases or servers.
- Not as quick as KVM (probably 80% as fast as KVM).

The NetBSD support has been upstreamed directly to the Intel HAXM repository at GitHub and packaged in pkgsrc.

6. Supported guest OSs

HAXM on NetBSD has been verified to successfully boot a number of guest Operating Systems, such as NetBSD/amd64 and (in alphabetical order):

- DarwinX86
- DragonflyBSD
- FREEDOS
- FreeBSD/i386
- Haiku
- KolibriOS
- Linux/amd64 (noapic)
- Minix3
- OpenBSD/amd64
- Plan9
- ToaruOS
- Windows 7 32-bit

A selection of OSs is still known to be not handled appropriately. An important target is NetBSD/i386 and (in alphabetical order):

- Android X86
- FreeBSD/amd64
- Icaros (AROS)
- OpenBSD/i386
- QNX
- ReactOS
- Solaris
- Windows in other versions

There are still issues specific to the NetBSD host, especially APIC and RDTSC calibration bugs and generic ones such as handling Guest CPU Registers and missing instructions in the embedded instruction emulator (needed for MMIO).

7. Conclusions

HAXM has been verified to be a portable cross-platform loadable kernel module in the NetBSD kernel context. Utility of the software has been verified with a number of Guest Operating Systems.

8. Acknowledgments

The HAXM community for helping to address generic bugs and openness for new platform support.

The NetBSD community for thorough testing and feedback in the porting process.

9. References

[1] The Hardware-Assisted Virtualization Challenge
http://blog.netbsd.org/tnf/entry/the_hardware_assisted_virtualization_challenge

[2] HAXM in pkgsrc
http://mail-index.netbsd.org/netbsd-users/2019/02/13/msg022207.html

[3] HAXM official repository
https://github.com/intel/haxm