“Honey, I shrunk the hypervisor”
The No EMUlation Hypervisor
October 25th, 2018
Motivation

- 49% CVEs for QEMU 2013 to 2018 are from emulated devices \(^1\)
- Modern hardware and KVM needs less software support
- Distribution support: virtio devices are available for all important cloud use-cases and widespread distribution support
- Fewer lines of code → fewer bugs, easier to audit

\(^1\) CVEs analysed from www.cvedetails.com based on whether description mentions emulated device
“found him!” - Hans Splinter

CC BY-ND 2.0
**NEMU**

- **No EMulation**
- QEMU based
- Exclusively focused on cloud-specific workloads
- Removes all features, platform and hardware emulation not required for the cloud
- Keeps the performance, stability and robustness of QEMU
NEMU Goals

- Small attack surface (Low complexity, low footprint)
- High performance (Comparable to QEMU)
- No hardware emulation
- UEFI firmware only
- Linux and modern Windows guests only
- x86-64 and aarch64 only
- *Hardware-Reduced* ACPI
- Hotplug: CPU, memory, NVDIMM and PCI
Initial Proof of Concept

Public Domain: Queensland State Archives #
Initial Proof Of Concept (May 2018)

- pc and q35 (i386), virt (ARM) machine types only
- Semi-manual code removal: features, platforms and virtual devices
- 75% code size reduction
- Binary size reduced from 12.5 MB to 4.3 MB
- Shared libraries reduced from 97 to 29
- Device model reduced from 236 to 66 devices
- Published on GitHub as “experiment/code-reduction”
Thinking smaller: new machine type

Public Domain: “nemo” Holly #
“i386/virt” machine type

- New machine type on top of QEMU 3.0.0: \textit{i386/virt}
- \textit{Hardware-Reduced} ACPI
- UEFI firmware only (OVMF)
- Minimized hardware emulation
- No legacy hardware support
- Minimized device model
- ACPI-based hotplug
- Integrated and extended CI
Hardware-Reduced ACPI

- **i386/virt** complies with the *Hardware-Reduced ACPI* specification
- Specifically designed for modern, legacy-free and UEFI-based platforms
- Significantly less complex ACPI core code
- Needs a [kernel patch](#) to support hotplug
UEFI Firmware

- One single virtual UEFI firmware: OVMF
- `i386/virt` support added to OVMF
- OVMF changes:
  - Replace fixed function ACPI timer with KVM clock + TSC
  - Don’t use CMOS to get memory details
  - Use KVM clock instead of emulated RTC
- Temporary [OVMF fork](http://www.01.org)
Minimized Hardware Emulation

- Minimal PCI host bridge emulation (pci-lite)
- No chipset-specific emulation (LPC, PCH, MCH)
- No ISA, SMBUS or RTC/CMOS emulation
- Clock and IRQ controller offloaded to KVM
- Virtual ACPI device for hotplug/shutdown/reset support
Hotplug

- CPU, memory, NVDIMM and PCI devices
- Not a common use case for cloud workloads
- Mostly needed for VM based containers support (Kata Containers)
- Purely ACPI based (even for PCI devices)
Continuous Integration

- NEMU Automatic Test System (NATS)
- Programmatic approach to QEMU functional testing
- Jenkins-based CI, each GitHub PR is tested across “pc”, “q35”, “aarch64/virt” and “i386/virt”
Status

- NEMU is already an open source project
- Temporary OVMF fork to support the new \textit{i386/virt} machine type
- The \textit{i386/virt} machine type boots UEFI-based Linux cloud workloads (Clear Linux, Ubuntu Xenial & Bionic)
- The \textit{i386/virt} machine type runs Kata Containers
- Published as “\texttt{topic/virt-x86}”
Thinking even bigger smaller:

CC-BY: “Angelfish with Anemone” Ratha Grimes
Shrinking the build

● Increased configurability of build:
  ○ No TCG dependency for ARM virt
  ○ Able to build without PC/Q35
  ○ More explicit CONFIG_ options

● Very minimal “virt only” default config: x86_64_virt-softmmu.mak
Automated code removal

- Automated removal scripts publishing automatically from pushes to “topic/virt-x86” branch.
- Builds
- Published as “experiment/automatic-removal”
QEMU vs NEMU: Code Size\(^1\) Reductions

<table>
<thead>
<tr>
<th></th>
<th>QEMU 3.0.0</th>
<th>NEMU</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1317932</td>
<td>276410</td>
<td>-79%</td>
</tr>
<tr>
<td>C header</td>
<td>257301</td>
<td>68863</td>
<td>-73%</td>
</tr>
<tr>
<td>Complexity</td>
<td>148831</td>
<td>42340</td>
<td>-71%</td>
</tr>
</tbody>
</table>

\(^1\) Measurements done with [https://github.com/boyter/scc](https://github.com/boyter/scc)
## QEMU vs NEMU: Device Model¹ reduction

<table>
<thead>
<tr>
<th></th>
<th>NEMU pc</th>
<th>NEMU virt</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of devices</td>
<td>221</td>
<td>45</td>
<td>-79%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>QEMU 3.0.0 pc</th>
<th>NEMU virt</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of devices</td>
<td>225</td>
<td>45</td>
<td>-80%</td>
</tr>
</tbody>
</table>

¹ Device model description built from QEMU’s info qdm command
Questions ?