NEVE: Nested Virtualization Extensions for ARM

Jin Tack Lim, Christoffer Dall, Shih-Wei Li, Jason Nieh, and Marc Zyngier*
Nested Virtualization

**Host Hypervisor**

**Guest Hypervisor**

**Hardware**

![Nested Virtualization Diagram]

- **Host Hypervisor**
- **Guest Hypervisor**
- **Hardware**
- **VM**
  - Nested VM
    - App
    - App
    - Kernel
  - Nested VM
    - App
    - App
    - Kernel
  - VM
    - App
    - App
    - Kernel
Nested Virtualization

- Run your own VM in public clouds

![Google Cloud Platform](https://www.google.com)
![Microsoft Azure](https://www.microsoft.com)
![ORACLE](https://www.oracle.com)
![ravello](https://www.ravello-systems.com)

**Diagram:**
- **Host Hypervisor**
- **Guest Hypervisor**
- **Nested VM**
  - Kernel
  - App
  - App
Nested Virtualization

- Run your own VM in public clouds
  - Google Cloud Platform
  - Microsoft Azure
  - ORACLE ravello
- Run OSes which have built-in hypervisors in a VM
arm
Key Problem

- No nested virtualization support on current hardware ARMv8.0
- Nested virtualization supported in future hardware ARMv8.3
- Nested virtualization performance on ARM is unknown
- ARM hardware virtualization support different from x86
Key Contributions

• Introduced paravirtualization for architecture evaluation

• Evaluated nested virtualization performance

• Proposed a new architecture extension, NEVE
  • NEVE improves performance up to 10x

• NEVE is included in the next ARM architecture, ARMv8.4
Evaluation Challenges

- No ARMv8.3 hardware, No idea about performance
  - ARMv8.0 is the latest hardware publicly available
- Long development cycles

Architecture Design → A few years.. → Hardware Release → Evaluation
Current Approaches

- Cycle-accurate simulators
  - Costly, too slow and lack of device support
- Simpler architecture models, e.g. ARM Fast Models
  - Provides only correct hardware functionality, not performance
Paravirtualization for Architecture Emulation

- Possible if existing hardware has instructions to mimic new architecture features
- Architectural features for virtualization often involve traps
Paravirtualization for Emulation of ARMv8.3

Instructions that do trap

Trap

Guest Hypervisor

Host Hypervisor

New Hardware

ARMv8.3

VM
Paravirtualization for Emulation of ARMv8.3

Armv8.0

VM

Guest Hypervisor

Instructions that don't trap

Host Hypervisor

Existing Hardware
Paravirtualization for Emulation of ARMv8.3

ARMv8.0

VM

Guest Hypervisor

Paravirtualized Instructions that do trap

Host Hypervisor

Trap

Existing Hardware
Benefits

• Makes possible to evaluate new architecture features with real workloads on real hardware

• Allows co-design and rapid prototyping of SW and architecture
  • Make development cycles short
Implementation

- Designed and implemented KVM/ARM Nested Virtualization
- First ARM hypervisor supporting nested virtualization
- Similar approach to Turtles [OSDI 2010] - KVM on x86
## Application Workloads

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernbench</td>
<td>Kernel compile</td>
<td>Netperf TCP RR</td>
<td>Network performance</td>
</tr>
<tr>
<td>Hackbench</td>
<td>Scheduler stress</td>
<td>Netperf TCP STREAM</td>
<td>Network performance</td>
</tr>
<tr>
<td>SPECjvm2008</td>
<td>Java Runtime</td>
<td>Netperf TCP MAERTS</td>
<td>Network performance</td>
</tr>
<tr>
<td>MySQL</td>
<td>Database management</td>
<td>Apache</td>
<td>Web server stress</td>
</tr>
<tr>
<td>Memcached</td>
<td>Key-Value store</td>
<td>Nginx</td>
<td>Web server stress</td>
</tr>
</tbody>
</table>
Experimental Setup

- ARM
  - APM X-Gene (ARMv8.0)
  - x86
    - Intel E5-2630 v3

- Native/VM/Nested VM Setup
  - 4-way SMP
  - Virtio (VM/nested VM)

- Software
  - KVM on KVM
  - PV ARMv8.3
Application Benchmarks

Normalized overhead (lower is better)

- ARMv8.3 VM
- ARMv8.3 Nested VM
- x86 VM
- x86 Nested VM

Benchmarks:
- Kernbench
- Hackbench
- SPECjvm2008
- TCP RR
- TCP STREAM
- TCP MAERTS
- Apache
- Nginx
- Memcached
- MySQL
Nested Virtualization

Why is it so slow on ARMv8.3?
ARM Virtualization Extensions

- EL0
  - App
  - App
  - App

- EL1
  - OS Kernel

- EL2
  - Hypervisor
ARM Virtualization Extensions

EL0
  App
  App
  App

EL1
  OS Kernel

EL2
  Hypervisor

EL1 System Registers
EL2 System Registers
ARM Virtualization Extensions

EL0: Applications

EL1: OS Kernel

EL2: Hypervisor

VM

TTBR0_EL1

TTBR0_EL2
ARM Virtualization Extensions

EL0
App  App  App

EL1
OS Kernel

EL2
Hypervisor

EL1 System Registers
EL2 System Registers
Nested Virtualization on ARM

- **EL0**: Applications
- **EL1**: OS Kernel
- **EL2**: Host Hypervisor

**VM**
- Nested VM
- App
- App
- App

**Host Hypervisor**

**Guest Hypervisor**

**Nested VM**

**OS Kernel**
Nested Virtualization on ARM

Diagram showing the layers of nested virtualization on ARM, with EL0, EL1, and EL2 levels, including App layers, Nested VM Exit and Entry, Guest Hypervisor, Host Hypervisor, and OS Kernel.
Nested VM Entry on ARM

- **VM**
  - **Nested VM**
    - **App**
    - **OS Kernel**
    - **Nested VM Exit**
    - **Nested VM Entry**

- **EL0**
  - **App**

- **EL1**
  - **Nested VM Exit**
  - **OS Kernel**
  - **Nested VM Entry**
  - **Guest Hypervisor**
  - **Host Hypervisor**

- **EL2**
  - **Trap**
  - **...**
  - **Trap**
  - **Trap**
  - **Trap**

- **Trap**
  - **...**
  - **Trap**
  - **Trap**
  - **Trap**
  - **Trap**
Exit Multiplication

• A single exit from the nested VM leads to lots of traps

• It slows down ARM nested VM performance badly

• x86 has this problem, but not bad as ARM
NEVE: NEsted Virtualization Extensions for ARM

- Supports unmodified guest hypervisors and OSes
- Improves performance of nested virtualization
- Provides two techniques to avoid traps based on register classification
Register Classification

- VM registers, which affect VM execution
- Hypervisor control registers, which affect hypervisor execution
VM Registers: Redirection to Memory

- NEVE redirects VM register access instructions to memory
- On nested VM entry, the host hypervisor can get VM register states from memory
VM Registers: Redirection to Memory

- NEVE redirects VM register access instructions to memory
- On nested VM entry, the host hypervisor can get VM register states from memory
Hypervisor Control Registers

- The hypervisor accesses them to control execution
  - EL2 registers
- Can’t apply the technique for VM registers
- Traps are handled by redirecting to EL1 registers in software
The hypervisor accesses them to control execution

- EL2 registers

Can’t apply the technique for VM registers

- Traps are handled by redirecting to EL1 registers in software

- Redirect in hardware instead!

Hypervisor Control Registers

<table>
<thead>
<tr>
<th>EL1</th>
<th>Guest Hypervisor</th>
<th>EL1 Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL2</td>
<td>Host Hypervisor</td>
<td>EL2 Registers</td>
</tr>
</tbody>
</table>
NEVE Evaluation

- NEVE is a new architecture extension, but no hardware
- Use paravirtualization for architecture evaluation
- Memory redirection emulation
  - Register access instructions -> load/store instructions
- Register redirection emulation
  - EL2 register access instructions -> EL1 register access instructions
## Application Workloads

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernbench</td>
<td>Kernel compile</td>
<td>Netperf TCP_RR</td>
<td>Network performance</td>
</tr>
<tr>
<td>Hackbench</td>
<td>Scheduler stress</td>
<td>Netperf TCP STREAM</td>
<td>Network performance</td>
</tr>
<tr>
<td>SPECjvm2008</td>
<td>Java Runtime</td>
<td>Netperf TCP MAERTS</td>
<td>Network performance</td>
</tr>
<tr>
<td>MySQL</td>
<td>Database management</td>
<td>Apache</td>
<td>Web server stress</td>
</tr>
<tr>
<td>Memcached</td>
<td>Key-Value store</td>
<td>Nginx</td>
<td>Web server stress</td>
</tr>
</tbody>
</table>
Normalized overhead
(lower is better)

- ARMv8.3 Nested VM
- NEVE Nested VM
- x86 Nested VM

Application Benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>ARMv8.3 Nested VM</th>
<th>NEVE Nested VM</th>
<th>x86 Nested VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernbench</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hackbench</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPECjvm2008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP RR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP STREAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP MAERTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nginx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memcached</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Introduced paravirtualization for architecture evaluation

• Built the first ARM hypervisor supporting nested virtualization

• Nested virtualization on ARMv8.3 performs poorly

• NEVE improved performance up to 10x

• NEVE is included in the next ARM architecture, ARMv8.4