Transplantation of VirtualBox to the NOVA microhypervisor

Norman Feske
<norman.feske@genode-labs.com>
Outline

1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
Outline

1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
Architecture overview

Transplantation of VirtualBox to the NOVA microhypervisor
Starting up a VM process

VM process

/dev/vboxdrv

open

kernel

/vboxdrv.ko

Transplantation of VirtualBox to the NOVA microhypervisor
VM process running

<table>
<thead>
<tr>
<th>root mode</th>
<th>non-root mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM process</td>
<td></td>
</tr>
<tr>
<td>load VMMR0</td>
<td></td>
</tr>
<tr>
<td>/dev/vboxdrv</td>
<td></td>
</tr>
<tr>
<td>kernel</td>
<td></td>
</tr>
<tr>
<td>vboxdrv.ko</td>
<td></td>
</tr>
<tr>
<td>VMMR0 / Hypervisor</td>
<td></td>
</tr>
</tbody>
</table>

Transplantation of VirtualBox to the NOVA microhypervisor
Entering the Guest OS

- root mode
- non-root mode

Guest OS

- VM process
- /dev/vboxdrv
- kernel
- vboxdrv.ko
- ioctl VM RUN
- world switch

Transplantation of VirtualBox to the NOVA microhypervisor
Risks for desktop virtualization

root mode

VM
process
/dev/vboxdrv
kernel
vboxdrv.ko
VMMR0 / Hypervisor
highly complex

Guest OS

non-root mode
Risks for desktop virtualization

- Root mode
- Non-root mode

Guest OS

- VM
- Process
- /dev/vboxdrv
- Kernel
- VBoxdrv.ko
- VMMR0 / Hypervisor

Access control?

Highly complex
Risks for desktop virtualization

- **root mode**
  - authorized to change the kernel
  - highly complex
  - access control?
  - /dev/vboxdrv

- **non-root mode**
  - highly complex

**Guest OS**

**VMMR0 / Hypervisor**

Transplantation of VirtualBox to the NOVA microhypervisor
Outline

1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
NOVA architecture

- NOVA Microhypervisor (9,000 SLOC)
- Resource management
- Apps
- Drivers
- Guest OS (non-root mode)
- Guest OS (root mode)

Transplantation of VirtualBox to the NOVA microhypervisor
Flow of a virtualization event

User-level VMM

Guest OS

NOVA

UTCB

VMCS

world switch
copy
Genode OS architecture

→ Application-specific TCB
Genode OS framework

Transplantation of VirtualBox to the NOVA microhypervisor
Genode combined with virtualization
Seoul VMM on top of Genode/NOVA

- Unmodified Guest OS
  - Kernel
  - virtual CPU
  - virtual RAM
  - virtual device
  - VMM

- Resource Multiplexer
- Device Driver
- Init

- Core

- NOVA Hypervisor

- User Mode
- Privileged Mode

Transplantation of VirtualBox to the NOVA microhypervisor
Device models and features of VirtualBox

+ Security of the Genode/NOVA architecture
Outline

1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
Identify the interesting parts

Entire VirtualBox code base
> 4 million lines of code (sloccount)

Narrowed to the interesting parts
> 2 million lines of code

<table>
<thead>
<tr>
<th>Location</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>src/VBox/VMM</td>
<td>src/recompiler</td>
</tr>
<tr>
<td>src/VBox/Main</td>
<td>src/libs/liblzf-3.4</td>
</tr>
<tr>
<td>src/VBox/Runtime</td>
<td>src/libs/liblzf-3.4/cs</td>
</tr>
<tr>
<td>src/VBox/Devices</td>
<td>src/libs/libxml2-2.6.31</td>
</tr>
<tr>
<td>src/VBox/Storage</td>
<td>src/libs/zlib-1.2.6</td>
</tr>
<tr>
<td>src/VBox/GuestHost</td>
<td>include/VBox</td>
</tr>
<tr>
<td>src/VBox/Disassembler</td>
<td>include/iprt</td>
</tr>
<tr>
<td>src/VBox/HostServices</td>
<td></td>
</tr>
</tbody>
</table>
Porting the VirtualBox Runtime to Genode

- Facilitate Genode’s existing infrastructure
  - 3rd-party software management tools
  - FreeBSD libc
  - Standard C++ library
  - POSIX threads

→ Most parts of the POSIX runtime could be reused
Enable subsystems one by one

- Guest memory (accessed by recompiler and device models)
  \textit{RAM, MMIO}
- I/O-port handling
- PGM, HWACCM, TM
- Device models, PDM, BIOS
- Host drivers
  - Using the “Basic front end”
  - Reimplement SDLConsole interface
A look inside a VM process

- Recompiler
- Execution Manager
- Hardware Acceleration
- Instruction Emulator
- VM Exit
- VM Enter

Transplantation of VirtualBox to the NOVA microhypervisor
Start with executing the recompiler only
Simple test scenario

Transplantation of VirtualBox to the NOVA microhypervisor
Increasing guest complexity

1. Custom-made Genode OS scenarios

2. Small Linux-based images (Tinycore, GRML)

3. Windows XP
Windows XP as a guest

- FB SDL
  - Framebuffer
  - Input

- VirtualBox
  - Init
  - Core
  - Linux

- LX Proxy FS
  - File system

VDI image

Transplantation of VirtualBox to the NOVA microhypervisor
Move scenario to NOVA

Transplantation of VirtualBox to the NOVA microhypervisor
Entering non-root mode

Transplantation of VirtualBox to the NOVA microhypervisor
Entering non-root mode

- VBox VM state ↔ NOVA UTCB state

- Virtualization of guest memory
  \( \text{(EPT faults)} \)

- Enter VT-x conservatively
  \( \text{(if protected mode and paging enabled)} \)

- Inject IRQs into recompiler

- Later: IRQ injection via NOVA into VT-X
Adding features

Additional drivers
- Networking

Guest tools
- Shared folders
- Host clock
- Mouse-pointer synchronization
Update to VirtualBox 4.3

- Basic front end no longer supported

- Use of main front end code to NOVA port
  - Custom console implementation
  - Shortcut XPCOM middleware
  → Support for using .vbox files
1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
Windows 7 running in VirtualBox directly on top of NOVA
Adaptation of VirtualBox to Genode/NOVA

Ported code
- 400,000 lines of code (sloccount)

New code
- 6,200 lines (sloccount)
  - hm, iommio, ioprt, mm, pdm, pgm, sup

Modifications of the original code
- 510 lines added
- 120 lines removed
Current state and outlook

- Usable performance, optimization ongoing
- Focused on VT-X, SVM not regularly tested
- Reduces TCB complexity to two orders of magnitude
- Useful for building appliances in high-security computing
- Stepping stone for using Genode as a general-purpose OS
1. VirtualBox

2. NOVA microhypervisor and Genode

3. Steps

4. Demo + Outlook

5. War stories
War stories

- Invalid guest state
- TLB consistency
- Interrupt handling
- Large files in shared folders
Thank you

Genode OS Framework
http://genode.org

Genode Labs GmbH
http://www.genode-labs.com

Source code at GitHub
http://github.com/genodelabs/genode