SHRINKING THE HYPERVERSOR ONE SUBSYSTEM AT A TIME
A Userspace Packet Switch for Virtual Machines

Julian Stecklina
OS Group, TU Dresden
jsteckli@os.inf.tu-dresden.de

VEE 2014, Salt Lake City
1. Motivation
2. Userspace Switch
3. Evaluation
4. Summary
01 Microkernels and TCB

Systems built on microkernels usually structured as multiserver systems with strong isolation between subsystems.

Applications only depend on subsystems they use.
01 Towards Monolithic Hypervisors
01 Towards Monolithic Hypervisors

- KVM
- Qemu
- virtio-net

Diagram showing layers of virtualization with KVM, Qemu, and virtio-net.
01 Towards Monolithic Hypervisors
01 Towards Monolithic Hypervisors
Attacks by malicious guest code are a serious concern.

Successful attacks on Qemu achieve unprivileged code execution.

Dangerous, but manageable.

Successful attacks on KVM achieve code execution in kernel mode.

Game over. You are here.
In a modern KVM installation the complete networking path is in the TCB of all applications on the host:

- (simple) instruction decoding,
- virtio-net device implementation,
- NIC driver.

Does it have to be?
1 Motivation

2 Userspace Switch

3 Evaluation

4 Summary
02 KVM Networking

```

VM A

VHOST

VHOST

VCPU B

VM Exit

Inject IRQ

notification

user

kernel

guest

```

---

TU Dresden  Userspace Packet Switch  slide 11 of 30
02 KVM Networking

VM A

VM Exit

Inject IRQ

VM B

kernel

user

guest

VCPU A

sv3

VCPU B

VM Exit Inject IRQ

notification
02 KVM Networking

- Kernel
- User
- Guest
- VM Exit
- Inject IRQ
- Notification
- CPU 1
- CPU 2
- CPU 3
- VM A
- VM B
- VCPU A
- VCPU B
- sv3
Userspace packet switch running as ordinary process on top of the host Linux:

- implements virtio-net,
- (no) packet memory management,
- NIC driver.

Every sv3 instance is a complete isolated networking subsystem.
KVM and vhost are loosely tied together by Qemu using `eventfds`. Qemu ties them together using `ioctl`.

- KVM can trigger `eventfds` on VM Exits.
- `eventfds` can be used to trigger IRQ injection in KVM.

Can use `eventfds` from userspace as well without using vhost.
Enhanced Qemu to support out-of-process PCI devices.

- Qemu connects to sv3 via `AF_LOCAL` socket.
- Qemu exchanges `fds` to establish shared memory.
- Qemu exchanges `eventfds` for
  - VM Exit notification,
  - IRQ injection.

sv3 implements the complete virtio-net logic and it feels like L4!
sv3 creates linear mappings of guest memory with `mmap`.

Packet data can be copied with a plain `memcpy`. No additional copies are necessary.

If no buffer space in the receiving VM is available, packet is dropped. No dynamic memory management for packets needed.
Userspace driver for Intel X520 10 GBit NIC using VFIO (requires IOMMU) supporting static offloads:

- TCP Segmentation Offload
- Large Receive Offload
- Checksum Offload

Virtio descriptors translated to HW descriptors allows for zero-copy send with all offloads.

Better reuse existing drivers next time . . .
sv3 is mostly single-threaded and lockless. Userspace RCU is used to synchronize adding and removing switch ports.

1. disable events on all virtio queues
2. disable HW IRQs
3. poll for work until queues empty
4. enable events/IRQs
5. poll a last time, if packet seen goto 1
6. block on `eventfd`

In overload scenarios, sv3 naturally operates in polling mode.
03 Resource Consumption

Processes are lightweight alternatives to driver VMs.

<table>
<thead>
<tr>
<th>sv3 NIC driver</th>
<th>&lt; 2 MiB + 14 MiB</th>
<th>sv3 total</th>
<th>&lt; 16 MiB</th>
</tr>
</thead>
</table>

smallest VM
netback VM

32 MiB [1]
128 MiB [1]

Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor, Colp et al., SOSP ’11
03 Evaluation System

- Intel Core i7 3770S (Ivy Bridge)
- C-states, HT and frequency scaling disabled
- 16 GiB RAM @ ~159 Gbit/s
- Host: Fedora 19 with Linux 3.10 (vanilla)
- Guest: Linux 3.10 (vanilla), 256 MiB RAM
- Qemu 1.5 (plus patches)
Notification to IRQ injection times for vhost vs. sv3 without any packet processing. Cost of additional trip to syscall layer and mode switch.
Latency measured using `netperf UDP RR`. 
03 VM-to-VM Bandwidth
03 VM-to-VM Bandwidth (cont’d)

![Graph showing CPU utilization vs. GBit/s for different configurations.

- vhost, no tso
- sv3, no tso
- sv3 only]
03 external-to-VM Bandwidth

![Graph showing CPU Utilization vs. GBit/s for different configurations: vhost, tso, sv3, tso, and sv3 only. The graph indicates a linear increase in CPU Utilization as the GBit/s increases.]
sv3 is an efficient lockless userspace packet switch for VMs running on (unmodified) Linux/KVM. Code: https://github.com/blitz/sv3.

Linux/KVM has all the mechanisms to make a microkernel-style design possible and efficient:

- rights transfer via AF_LOCAL sockets (capabilities)
- efficient notifications via eventfds
- drivers in userspace via VFIO using eventfds for IRQs
- tying eventfds to VM exits / IRQ injection
- address space switch cost not a factor in performance

Few reasons to write new systems functionality in kernel mode.

Questions?
04 external-to-VM Bandwidth

![Graph showing CPU Utilization vs. GBits/s for 'sv3, no tso' and 'sv3 only' configurations.](image)