Virtio 1 - why do it?
And - are we there yet?

2015

Michael S. Tsirkin
Red Hat

Uses material from
https://lwn.net/Kernel/LDD3/
Gcompris, tuxpaint
Distributed under the
Creative commons license.
Lots of work ...
Virtio 1: update

- Documented assumptions
- More Robust
- More Extendable
## Conformance statements

<table>
<thead>
<tr>
<th>Virtio 0.9</th>
<th>Virtio 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>- DRIVER_OK status bit is set.</td>
<td>The driver <strong>MUST NOT</strong> notify the device before setting DRIVER_OK.</td>
</tr>
<tr>
<td>- The device can now be used.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><code>drv→probe(dev);</code></td>
<td><code>drv→probe(dev);</code></td>
</tr>
<tr>
<td><code>netif_carrier_on(dev)</code></td>
<td><code>add_status(dev, DRIVER_OK);</code></td>
</tr>
<tr>
<td></td>
<td><code>netif_carrier_on(dev)</code></td>
</tr>
<tr>
<td><code>add_status(dev, DRIVER_OK);</code></td>
<td></td>
</tr>
</tbody>
</table>
Virtio 0.9: inflate

```
0.......................................31
FF FF  00  00

+1

0.......................................31
00  00  01  00
```

```
FF FF  01  00
```

 DRIVER
Virtio 1.0: inflate

00 00 01 00

+1

00 00 01 00

DRIVER
Generation counter

0

00000000

00000000

DRIVER

1

+1

00000000

00000000

00000001

00000001
Memory map

0.9

- COMMON
  - FEATURES
  - QUEUE
  - STATUS
  - ISR

- DEVICE SPECIFIC

1.0

- COMMON
  - FEATURES
  - QUEUE
  - STATUS
  - ISR

- DEVICE SPECIFIC

- CAPABILITY LIST
- IO BAR
- MEMORY BAR
- VIRTIO CAPABILITY #1
- VIRTIO CAPABILITY #2
## Virtio 0.9: Port IO vs Memory

<table>
<thead>
<tr>
<th></th>
<th>Port IO</th>
<th>MM IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86 decode: address</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>x86 decode: data</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Fast on x86</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>32/64 bit</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Page tables</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Required by PCI Express</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Fast MMIO
avoid need to decode data

<table>
<thead>
<tr>
<th>VQ NUMBER</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.........15</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>VQ NUMBER</td>
</tr>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>NOTIFY</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.........15</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>VQ NUMBER</td>
<td>DATA</td>
</tr>
<tr>
<td></td>
<td>NOTIFY</td>
<td>IGNORED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>VQ NUMBER</th>
<th>NOTIFY</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.........15</td>
<td>16..........31</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>VQ NUMBER</td>
<td>NOTIFY</td>
<td>IGNORED</td>
</tr>
</tbody>
</table>

1.0
Virtio 1: Access times on KVM x86: Cycles per access (lower is better)
## Virtio 1: Port IO vs Memory

<table>
<thead>
<tr>
<th></th>
<th>Port IO</th>
<th>MM IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86 decode: address</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Fast on x86</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>32/64 bit</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Page tables</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Required by PCI Express</td>
<td>✗</td>
<td>✔️</td>
</tr>
</tbody>
</table>
# Memory Region Aliases

<table>
<thead>
<tr>
<th>Capability List</th>
<th>IO BAR</th>
<th>Memory Bar</th>
<th>VirtIO Config Capability</th>
<th>VirtIO Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Queue Notify</td>
<td>Queue Notify</td>
<td>Queue Notify</td>
</tr>
</tbody>
</table>

VirtQueue
soft mac

Ethernet MAC

VirtQueue

DRIVER

0.9

1.0
Virtio feature negotiation

| 0 | 1 | 1 | -|- |
|---|---|---|---|
| DRIVER FEATURES | DEVICE FEATURES |

| 0 | 0 | 1 | -|- |
|---|---|---|---|
| DRIVER FEATURES |

Defaults must be maintained forever!
Virtio 1: Error handling

• DRIVER: set features
• DRIVER: set FEATURES_OK bit

• DEVICE: check features
• DEVICE: clear FEATURES_OK on error

• DRIVER: check FEATURES_OK bit
• DRIVER: fail gracefully if not set
Error handling: Virtio 0.9

• Can't recover from device errors
• Not very useful?
• Just stop guest.
Client crash or restart need not cause guest crash!
DEVICE_NEEDS_RESET

Read STATUS;
Detect: NEEDS_RESET set

Write STATUS=0
Will reset device

Reconfigure device.
Write STATUS=DRIVER_OK
Restart operation.
Compatibility

Transitional Device & Driver

Legacy Driver

Legacy Device
Are we there yet?

GUEST

BIOS

VHOST USER

KVM

VHOST
What to expect?

- **Current:** Virtio-v1.0-cs03
- **Next bugfix:** Virtio-v1.0-cs04
  - Virtio-blk: writeback / writethrough control
  - More update guidance
- **Next feature:** Virtio-v1.1-cs01
  - Virtio-input
  - Virtio-gpu
  - Virtio-vsock
TX: Interrupt avoidance
TX: Interrupt coalescing
Pass-through for nested virt

- Memory mapped: use page tables
- IOMMU: translate and protect guest memory

Virtio Net (on host)
Virtio as PCI Express device

- Uses memory mapped IO support
- Multi-root for NUMA
- Native hotplug
- Advanced Error Reporting
Summary

• Why do it?
  – Improved robustness for virtual devices

• Are we there yet?
  – Yes!
  – And there's more to come.
Thank you!
Virtio 0.9: Port IO versus memory on KVM x86: cycles per access (lower is better)
OASIS Virtio TC

Virtio 1.0

- PCI
- MMIO (ARM)
- CCW (PPC)
Virtio 1.0

• Virtio PCI:
  – Replace Port IO with Memory mapped IO
  – PCI Express (hotplug, AER, multi-root, SRIOV)
  – Infinite features

• Reduced memory requirements
• Fixed endianness
• Compatibility
Port IO: outl

VM Exit

VM Exit

REASON

QUALIFICATION

STATE
Memory mapped IO: writel

89

MOV

3E

(%EDI)

%RSI

PTE

VALID?

VM Exit

REASON

GUEST ADDRESS

RIP
Fast MMIO

MOV (%EDI)

notify

VQ#

PTE

VALID?

%RSI

VM Exit

REASON

GUEST ADDRESS
Multiple interfaces

- CAPABILITY LIST
- IO BAR
- MEMORY BAR
- VIRTIO CAPABILITY #1
- VIRTIO CAPABILITY #2
Memory requirements

0.9

VQ -> desc | avail | used

1.0

VQ -> desc | avail | used
features

0.0

0.................................31

0 1 1 -|

DEVICE FEATURES

DRIVER

0 1 0 1

DRIVER FEATURES

1.0

SEL 1 2 3 4 ..... 

0... .... .... .... .... ..... 

DRIVER

.... .... .... ...

STATUS = FEATURES_OK
Endianness

Virtio 0.9

intel
Virtio LE

Device LE

PPC
Virtio BE

Device BE

Virtio 1.0

Device

Virtio LE

Device

Device
compatibility

0.9

Driver

Device

1.0

Driver

Device

compatibility
Packet layout

Virtio 0.9

INDIRECT → next → header

Virtio 1.0

header → header
Packet layout: transactions per sec (higher is better)
More: virtio 1.0 versus 0.9.5

- Virtio 9p
- Virtio blk: WCE
- Virtio-net Multiqueue
- Virtio-net dynamic offloads
- Already upstream (based on spec draft)
vhost updates

- Vhost scsi
- Vhost-net  zero copy transmit
- No need for driver changes
Kvm networking

- Openvswitch – if time allows
- Ethernet bridge
Bridge FDB

London
Heathrow
Paris
CDG

uplink

London
Paris
Heathrow
CDG

London
Paris
Flood: DOS potential

- London
- Heathrow
- Paris
- CDG

uplink

Bangkok

Heathrow

CDG

London

Paris
Disable flood

London  Heathrow
Paris    CDG

uplink

London  Bangkok  Paris

Heathrow  CDG

London  Paris
softmac

- `ifconfig eth0 hw ether 00:12:23:45:67:89`
Using softmac/non promiscuous
Work in progress

- ELVIS (vhost blk/vhost net)
- Virgl
- Vhost-net performance
RX latency

NIC
HOST
VHOST
VM
Fast rx: transactions per sec (higher is better)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit</td>
<td>331668</td>
<td></td>
</tr>
<tr>
<td>Miss</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>
Vhost-net threading

- **tap**
- **VHOST**
- **VM**
- **NIC**
- **VHOST**
- **VM**
Vhost-net thread pool

tap
VM
NIC
VM

WQ
VHOST
VHOST
threading: UDP RR
transactions/sec (higher is better)
threading: TCP STREAM transactions/sec (higher is better)
summary

- Performance
- Manageability
- Security
Questions?
OVS: flow match

 kernel

userspace

PACKET FLOW

192.68.0.1 22

192.68.0.1 12865

VM

22

12865

OVS-VSWITCHD
OVS: wildcard match

PACKET FLOW

22
12865

192.68.0.1 *

VM

kernel

userspace

OVS-VSWITCHD
Wilcard: netperf CRR (higher is better)