Virtio-loopback: a common device interface between Virt-AGL and Non-Virt AGL

Michele Paolino
2022-10-19
m.paolino@virtualopensystems.com
www.virtualopensystems.com
The “Common device interface between Virtualized and non virtualized AGL” project is composed by two main tasks:

- Virtio-loopback
- Touchscreen with sensitivity support

This presentation will go through the current status of both.
Presentation index

- Objectives
- Requirements
- Design overview
- Benefits
- Current Status
- Virtio-loopback components
  - Kernel space
  - User space
- Next steps

virtio-loopback

touchscreen-sensitivity

- Requirements
- Current Status
- Next steps
Build a virtio based hardware abstraction layer that:

- Enables applications portable execution on:
  - Native and virtual machines environments
  - Several hypervisors
- Provides a standardized solution to build userspace drivers for both native and virtualized environments
Design overview

Two new components have been designed and developed:

- A user space application (virtio-loopback-adapter)
- A kernel driver (virtio-loopback)
virtio-loopback

Kernel component composed by

- Char device (interface with virtio-loopback adapter used to set up control plane)
- Virtio transport (used as interface between the virtio driver and the vhost-user device)
Userspace application that build the connection between the device and the virtio kernel driver

- It implements vhost-user protocol to communicate with the vhost-user device
- Uses syscalls (write/ioctl) to interact with virtio-loopback
Benefits

- Existing user-space implementations can be reused
- Hypervisors that support virtio/vhost-user standards are fully compliant
- Data (vrings) exposed by the virtio driver in kernel space and directly mmap()ed by the device in user space
  - no copies, higher performance!
- Host and user space components are fully compliant with virtio and vhost-user open standards
  - virtio/vhost-user guarantee openness and stability
Alpha release publicly available for review, test and download: https://git.virtualopensystems.com/virtio-loopback/

- Support for both x86 and Arm processors (tested on KVM virtual machines and R-Car M3 with AGL needlefish)
- Runs with both C and RUST vhost-user-rng implementations
virtio-loopback alpha release demo
Video demonstration publicly available!

As of today, a the alpha release supporting RNG has been delivered and reviewed by part of the community. Next steps:

- Gather and address further comments from AGL community
- Add support for input and block devices
- Prepare a Request for Comments (RFC) for relevant communities
  - To be evaluated proposals to kernel, QEMU, rust-vmm
- Final version before AGL ALS
The target solution agreed with EG-IVI is based on the MatrixOrbital HTT70A R1.0.0 device

- Supports different levels of touchscreen sensitivity
- HDMI touchscreen, open source driver
  - Can be used with different hardware platforms

```c
int set_sensitivity(hid_device *handle, int sensitivity)
{
  unsigned char buf[256];
  buf[0] = REPORT_MXT_SENSITIVITY;
  buf[1] = sensitivity;
  int res = hid_send_feature_report(handle, buf, 2);
  if (res < 0) {
    return 0;
  }
  return 1;
}
```
The touchscreen device has been tested on AGL needlefish (Rcar M3)
Code changes - HIDRAW kernel support

```
mpaol@vosys-s9:~/shared/test/meta-agl/meta-agl-bsp/meta-rcar-gen3/recipes-kernel$ git show
commit 0ef116f65cd867922fe102b5a7ad1582e118b685
Author: Michele Paolino <m.paolino@virtualopensystems.com>
Date:   Fri Sep 23 10:24:02 2022 +0200

Add kernel support to HIDRAW for R-Car gen3

Signed-off-by: Michele Paolino <m.paolino@virtualopensystems.com>
Change-Id: I0260888954bc96b7c4819c83e8ccc65667ae0e0

new file mode 100644
index 000000..bbd271a
--- /dev/null
+++ b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/files/hidraw.cfg
@@ -0,0 +1 @@
+CONFIG_HIDRAW=y

index 78996bf..0bbd346 100644
--- a/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/renesas_%.bbappend
+++ b/meta-agl-bsp/meta-rcar-gen3/recipes-kernel/linux/renesas_%.bbappend
@@ -10,6 +10,7 @@ SRC_URI:append = " \
   AGL_KCONFIG_FRAGMENTS += "namespace_fix.cfg"
   AGL_KCONFIG_FRAGMENTS += "Set_GOV_PERFORMANCE.cfg"
   AGL_KCONFIG_FRAGMENTS += "vivid.cfg"
+AGL_KCONFIG_FRAGMENTS += "hidraw.cfg"

# For Xen
AGL_KCONFIG_FRAGMENTS += " \
```
touchscreen next steps

As of today, touchscreen has been found and tested on the RCar M3. Next steps:

- Integrate it with vhost-user-input and virtio-loopback (AGL ALS)