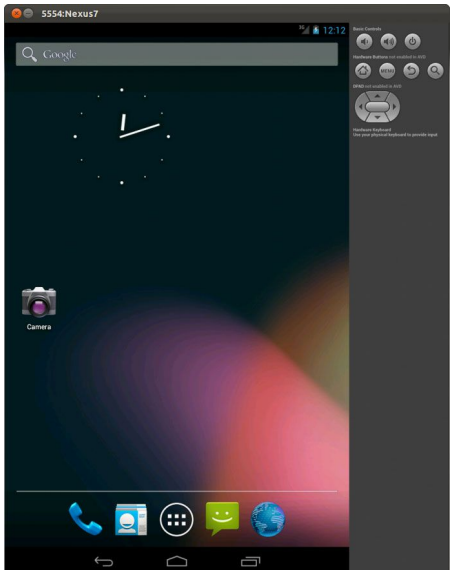
The Android Emulator and Upstream QEMU

Christoffer Dall
<cdall@linaro.org>



LEADING
COLLABORATION
IN THE ARM
ECOSYSTEM

The Original Android Emulator: qemu1

- Original Android emulator fork of QEMU 0.8.2 (2006)
  [Android Donut SDK repo, 2009]

- Difference from QEMU 0.8.2
  1,435 files changed 372,311 insertions(+) 176,000 deletions(-)
  (includes lots of backports from upstream QEMU)

- Goldfish board (goldfish fb, emulated nand+mmc, goldfish pipe, skinning, adb, console)

The Original Android Emulator

Version	non-android diffstat
Donut	584 files changed, 95520 insertions(+), 166845 deletions(-)
Eclair	724 files changed, 165429 insertions(+), 175396 deletions(-)
Froyo	764 files changed, 177414 insertions(+), 175254 deletions(-)
Gingerbread	791 files changed, 175267 insertions(+), 176171 deletions(-)
Ice Cream Sandwich	857 files changed, 224918 insertions(+), 165262 deletions(-)
Jelly Bean	873 files changed, 243679 insertions(+), 160910 deletions(-)
Kit Kat	877 files changed, 244642 insertions(+), 160922 deletions(-)
Lollipop	926 files changed, 267241 insertions(+), 203292 deletions(-)

QEMU History

Project	Version	Date	Notable changes/features
QEMU	0.8.2	Jul 2006	arm, i386, mips, ppc, sh4, sparc
QEMU	0.10.0	Mar 2009	TCG introduced
QEMU	1.0	Dec 2011	First of maj.minor releases
QEMU	2.0	Apr 2014	ARMv8 linux-user support
QEMU	2.1	Aug 2014	ARMv8 System Emulation support
QEMU	**2.2**	**Dec 2014**	**Base version used for Linaro ranchu work**
QEMU	2.7	Sep 2016	TCG speed increases of ~20%
QEMU	2.9	Apr 2017	MTTCG support for ARM
QEMU	2.10	Aug 2017	Latest Release

The New Android Emulator: qemu2

- Initial work done by Linaro for AArch64 support based on QEMU v2.2
- New board definition Ranchu based on ‘virt’ board
- Support for ADB and emulator commands using existing QEMU subsystems
- Forked by Google and maintained by Google
- Upstream QEMU releases are merged at regular intervals

The Ranchu Board

- Initial idea was to use existing standards
- Example: VirtIO instead of the pipe
- Virtio-based network, block, and console
- Goldfish devices (legacy support):
  - Pipe (host-guest transport)
  - Simple frame buffer
  - Battery status
  - Audio
  - Event device (virtual keyboard, mouse, touchscreen)
  - Sync device
  - ...
- Not upstream

Google changes to Ranchu board

- Glue Layer
  - cpp based
  - Device properties and setup (skinning etc.)
  - Sensor manipulation (feeding GPS and accelerometer inputs)
  - Thread handling changes
  - also a wrapper around real QEMU
- Graphics (GL pass-through)
  - Uses the pipe for data transfer
  - Multi-platform (i.e. Windows and Mac)
- Tracing and Metrics
- Build system
- Not upstream

Suggestions for future work

- Replace the pipe with VirtIO
- Virtio-vsock for low bandwidth zeroconf connection
- Virtio based accelerated graphics with Windows and Mac support
- Upstreaming to QEMU
- Have drivers be non-goldfish specific in the kernel
- KVM Support on AArch64

Other work

- Generic device overlay for AOSP which allows building Mesa-based android images with more or less vanilla kernels and upstream QEMU using virtio-GPU. Maintained by Rob Herring (Linaro).
  https://github.com/robherring/generic_device/wiki

- Patches porting MacOSx Hypervisor.framework from the downstream Android Emulator code.

Thank You

For further information: www.linaro.org

Android Emulator Kernel

September, 2017

Jin Qian <jinqian@google.com>

Terminologies

- Goldfish
  - virtual board based on qemu 1.10 (qemu1, a.k.a classic android emulator)
  - goldfish virtual devices (MTD block/nand/mmc) and corresponding virtual drivers
  - pdev_bus for device enumeration
  - <arch>_emu_defconfig for goldfish virtual board
- Ranchu
  - virtual board based on qemu 2.8 (qemu2, a.k.a ranchu android emulator)
  - virtio devices and drivers (block, net, console, pci)
  - DT/ACPI based device enumeration
  - <arch>_ranchu_defconfig for ranchu virtual board
  - goldfish_pipe (v2), goldfish_sync, goldfish_dma for perf critical path.
  - kernel branch and driver names are still using goldfish for legacy reasons.

Git repositories

- AOSP kernel/common.git
  - downstream of Linux
  - maintained by the Android team, contains Android-specific but non-vendor-specific changes, that are not yet in upstream Linux.
  - Branch names following android-<version>
- AOSP kernel/goldfish.git
  - forked from kernel/common.git, and used to contain emulator-specific changes related to goldfish and ranchu.
  - Branch names following android-goldfish-<version>

Branches

- android-goldfish-3.4 (qemu1 only, deprecated)
- android-goldfish-3.10 (being deprecated)
  - emu_defconfig - qemu1
  - ranchu_defconfig - qemu2
- android-goldfish-3.18 (current active branch)
  - ranchu/qemu2 only
  - Shipped with Android Oreo Release
  - MIPS maintained by imgtec
- android-4.4+ (future branches)
  - ranchu/qemu2 only
  - emulator kernel development happens in common.git
    - DT/ACPI driven, no board code
    - most goldfish driver patches sent upstream

Current work

- goldfish_pipe - v2 (Yurii Zubrytskyi <zyy@google.com>)
  - qemu_pipe virtual device as a generic interface for fast multiplexed guest <-> host communication
  - rewrite of old pipe driver to reduce guest/host transitions and lower latency
  - shared cmd buffer between host and guest
    - contains physically contiguous chunks of guest memory
  - over 2x improvement in adb push performance
- goldfish_dma (Lingfeng Yang <lfy@google.com>)
  - extension of pipe virtual device for high bandwidth use cases (e.g. 60fps video playback).
  - alloc and map guest physical memory to host in pipe channel.
  - guest writes to mmap-ed pipe channel are immediately visible to host.

Current work

- goldfish_sync (Lingfeng Yang <lfy@google.com>)
  - virtual device to synchronize drawing events for gpu emulation
    - host signals timeline/fence events (creation, inc, deletion) via guest interrupts
    - guest queues up work to wait for completion from host
  - low latency for host to talk with Android sync framework in guest
- cts/vts compliance
  - goldfish_fwdata.c - add fstab in device tree for Treble.
  - wifi and miscellaneous bug fixes only found by running android emulator.

Challenges

- VM snapshot
  - save/restore entire guest dram takes a long time (large dram size, slow storage)
  - lazily restoring guest pages when accessed has latency issue
  - ideas
    - async save (qemu live migration?)
    - restore hot pages first, then rest of pages in background
- Portability
  - requires modern VT features, lacks support for windows + AMD cpu, incompatible with hyper-V
  - qemu + TCG slow for ARM guest, no ARM hypervisor support
  - no nested virtualization support
  - ideas
    - paravirtualization

THANK YOU

Q & A