Android-x86 status update
from lead developer Chih-Wei Huang

Graphics stack evolution

presented by Mauro Rossi (maurossi)
Android-x86 status update

- Android-x86 project origins
- Android-x86 as seen in analytics
- Porting proceedings
- How to contribute to add new supported Hardware
- Stable and development release status
- Future Plans
Android-x86: what is it?

It is an Android Board Support Package*

For generic x86 platforms, including Desktop, Laptops

Effective Desktop OS alternative, with “Mouse Click as a Touch”

Completely Open Source and tracked as Linux OS in DistroWatch

Started in 2009 by Chih-Wei Huang and Yi Sun (beyouunn)

*https://en.wikipedia.org/wiki/Board_support_package

android-x86.org
3 million downloads reached and users still growing
### Android-x86 in analytics (2/2)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>1,250,778</td>
</tr>
<tr>
<td>Sessions</td>
<td>5,355,836</td>
</tr>
<tr>
<td>Screen Views</td>
<td>82,563,585</td>
</tr>
<tr>
<td>Screens / Session</td>
<td>15.42</td>
</tr>
<tr>
<td>Avg. Session Duration</td>
<td>00:18:46</td>
</tr>
<tr>
<td>% New Sessions</td>
<td>23.56%</td>
</tr>
</tbody>
</table>

- **Returning Users**: 75.7%
- **New Users**: 24.3%
Porting proceedings

• Prepare the codebase manifest xml
  • AOSP git projects
  • Modified AOSP git projects
    • device/generic/x86
    • device/generic/common (BoardConfig.mk, init.sh)
    • frameworks/base (SurfaceFlinger)
    • hardware/libhardware{*}
    • system/core
    • kernel
  • Android-x86 git projects
    • bootable/newinstaller
    • external/drm_gralloc
    • external/libdrm (update versions)
    • external/mesa (updated versions)
    • external/llvm (updated versions)
“Agile by necessity”

Apply android-x86 patches (git rebase) to AOSP

Try to build and fix compiling errors

Try to boot, identify and fix problems
  • Display is one the most critical aspects
Mostly manual/visual testing
Android CTS for EGL/GLES

We need effortless CI tools for Build, Boot and GUI tests

Most probably never, let’s start from CI
Overview of /etc/init.sh

Key part of android-x86 runtime drivers configuration
Based on sh script, with specialized init HAL functions

Adding new hardware support requires:
• linux kernel support (pre-requisite)
• change init_hal_* - e.g. init_hal_bluetooth() for Bluetooth

Patches from community are welcome to extend HW support
nougat-x86 features

- Android 7.1.2_r36 based
- 32-bit and 64-bit systems support
- Kernel 4.9, KMS enabled
- 3D Hardware acceleration (Intel/AMD/Nvidia graphics chipsets)
- Live CD & hard disk installation, **Auto Update feature**
- Hybrid iso image and EFI image
- Mouse, Wifi, Ethernet, Audio, Bluetooth and Sensors
- Drivers Auto-detect at init
- Virtual Machine installation Qemu, Virtualbox and VMware Player supported
oreo-x86 features

- Android 8.1.0_r46 based
- 32-bit and 64-bit systems support
- Kernel 4.17 wip, KMS enabled
- 3D Hardware acceleration (Intel/AMD/Nvidia graphics chipsets), HWC wip, Vulkan wip
- Live CD & Hard Disk installation, Auto Update feature
- Hybrid iso image and EFI image
- Mouse, Wifi, Ethernet, Audio, Bluetooth and Sensors
- Drivers Auto-detect at init
- Virtual Machine installation supported Qemu, Virtualbox and VMware Player
- Multimedia: Hardware accelerated codecs, HDMI audio
- LineageOS integration

android-x86.org
# Android-x86 porting status

<table>
<thead>
<tr>
<th>Feature</th>
<th>nougat-x86</th>
<th>oreo-x86</th>
<th>pie-x86</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>HW Acceleration</strong></td>
<td>OK</td>
<td>OK*</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Audio</strong></td>
<td>OK</td>
<td>OK*</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Wifi</strong></td>
<td>OK</td>
<td>OK*</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Bluetooth</strong></td>
<td>OK</td>
<td>OK*</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Sensors</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Camera</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Touchscreen</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Backlight</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
<tr>
<td><strong>Suspend/Resume</strong></td>
<td>OK</td>
<td>OK</td>
<td>Starting soon</td>
</tr>
</tbody>
</table>

*community is asking for additional HW support 3D, Audio, Wifi, Bluetooth*
Embrace the CI culture, one step at a time

Release HWC + Vulkan enabled iso (atomic & working drivers only)

Improve hardware codec support (NV12, YV12)

Porting Android P/9.0 release to pie-x86
Developers/contributors:

Many thanks to contributors to android-x86 and devs ML:

Chih-Wei Huang - codebase
Jaap Jan Meijer - cyanogenmod branches, LineageOS integration
Jon West - bliss-x86 developer, contributing to pie-x86
Michael Goffioul - multimedia codec acceleration
Paulo Sergio Tartaglia - nouveau and radeonsi first build and testing
Mauro Rossi - android-x86 mesa lifecycle, kernel RC forward-porting and testing
Emil Velikov - mesa release manager, supporting android-x86 developers Mailing List
Rob Herring - gbm_gralloc, virgl; Professor Chen - virgl
Robert Foss - drm_hwcomposer maintainer
Tapani Pälli - providing insights about android-ia/Intel patches
Tomasz Figa - prime fd fixes in mesa egl/android
Qiang Yu - help with amdgpu Alex Deucher guidance and review of RGBA patches

Join here: https://groups.google.com/forum/#!forum/android-x86-devel

android-x86.org
Call for enthusiasts

Ideal platform for HW not supported by new Operating Systems

In many cases Games & Controllers work Out-of-the-Box

Anybody with linux experience and passion can contribute, for example with improvements to alsa audio configuration, or to add support for touch screens, x86 tablet sensors, keybuttons

For enthusiasts and supporters, come visit us at:

https://groups.google.com/forum/#!forum/android-x86
Graphics stack evolution

- Overview of OSS graphic stack in android-x86
- Legacy drm Gralloc: supported HW
- drm HWComposer + gbm Gralloc HAL: test results
- AMDGPU support in android-x86
- Vulkan HAL: anv, radv build integration and testing
- Overview of outstanding work required/Next Steps

Live Demo with Vulkan in oreo-x86 with HWC1

Mauro Rossi
Legacy vs HWC reference

**Application**
- Apps

**Framework**
- SurfaceFlinger
- SurfaceFlinger
  - HWC

**HAL/ User space**
- no hwcomposer
- drm_hwc
- drm_gralloc
- mesa
- libdrm
- gbm_gralloc
- mesa
- libdrm

**Kernel**
- drm/kms
  - drm/kms

**Hardware**
- GPU
  - GPU
<table>
<thead>
<tr>
<th>Application</th>
<th>Framework</th>
<th>HAL/ User space</th>
<th>Kernel</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apps</td>
<td>SurfaceFlinger</td>
<td>no hwcomposer</td>
<td>drm/kms</td>
<td>GPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>drm_gralloc</td>
<td>mesa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>libdrm</td>
<td>libdrm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SurfaceFlinger</td>
<td>drm_hwc</td>
<td>gbm_gralloc</td>
<td>GPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mesa</td>
<td>mesa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>libdrm</td>
<td>libdrm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SurfaceFlinger</td>
<td>IA-Hardware-Composer</td>
<td>minibm_g</td>
<td>GPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>gbm_gralloc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mesa</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>libdrm</td>
<td></td>
</tr>
<tr>
<td>Gralloc HAL</td>
<td>Gralloc API</td>
<td>GEM / flink names</td>
<td>prime_fd</td>
<td>binderization</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>-------------------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>drm_gralloc</td>
<td>0</td>
<td>Y</td>
<td>uncomplete untested</td>
<td>N</td>
</tr>
<tr>
<td>gbm_gralloc</td>
<td>0</td>
<td>N/A</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>minigbm</td>
<td>0, 1</td>
<td>N/A</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Supported GPU/vGPU drivers [Mesa changes]
- Intel (i915, i965) [null pointer guards to avoid Playstore crashes]
- AMD (r300, r600, radeonsi, amdgpu) [allow non dma texture images]
- Nvidia (nouveau) [mutex locking patches required]
- Vmware (vmwgfx)
- QEMU (virgl)

Not supported
- Intel Atom with PowerVR IGP (no open source driver)

SwifShader used as fallback for unsupported HW
Contributions to drm Gralloc codebase:

- BGRA pixel format for chipsets lacking RGBA format
- Nouveau backend: support and testing for Kepler, Maxwell, Pascal
- Pipe backend: testing with amdgpu (SI, CIK, VI, AI)
- Pipe backend: fix gralloc sw usage flags handling (linear bo)
- Pipe backend: support changes in Mesa 18.2 gallium pipe/winsys
- Mesa 18.2: fixes to egl/android to drm gralloc path

Mesa egl/android periodic testing to verify no regression in drm Gralloc
drm HWComposer + gbm Gralloc path enablers:

- atomic display framework (driver dependent)
- explicit fencing new sync API (Gustavo Padovan - kernel 4.10)
- drm_hwcomposer (Sean Paul, Zach Reizner and Robert Foss)
- gbm gralloc - prime fd and “binderization” (Rob Herring)
- gralloc handle struct harmonized in libdrm (Rob Herring & Robert Foss)
- mesa support for dma-bufs prime_fd in egl/android (Tomasz Figa)
First build results

nouveau, amdgpu (amd dc):
black screen and top left white cursor

i965:
bootanimation crash-loop
E hwc-drm-device: Could not find a suitable encoder/crtc for display 2
E hwc-drm-device: Failed CreateDisplayPipe 56 with -19
E hwcomposer-drm: Can't initialize Drm object -19

---

inspired by Jim Bish commit in android-ia branch
beginning of crash

F libc    : Fatal signal 11 (SIGSEGV), code 1, ... in (surfaceflinger)
F DEBUG   : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
F DEBUG   : Revision: '0'
F DEBUG   : ABI: 'x86 64'
F DEBUG   : pid: 2853, tid: 2853, name: /system/bin/surfaceflinger <<<
F DEBUG   : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x8
F DEBUG   : Cause: null pointer dereference
F DEBUG   :
F DEBUG   : backtrace:
F DEBUG   : #00 pc 000000000001ebf0 \\
/system/vendor/lib64/hw/hwcomposer.drm.so (android::DrmEncoder::crtc() const)

addr2line output
addr2line -Cf symbols/system/vendor/lib64/hw/hwcomposer.drm.so 000000000001ebf0
android::DrmEncoder::crtc() const
external/drm_hwcomposer/drmencoder.cpp:39 line is [ return crtc_;]
Enumerates displays

Regardless of their state

First connector became always the primary

Removed
Possible explanation from Android source documentation of HWC

- It is assumed that there is exactly one primary display; that is, that there is one physical display that will be hotplugged immediately during the initialization.
- In addition to the primary display, exactly one external display may be hotplugged during normal operation of the device. (Not yet implemented in drm_hwc)
nouveau:
● Google Play crash at Apps search with nouveau
● Cursor tiling artifacts (buffer stride issues)
● Gralloc Usage Read/Write often swap buffer issues (No text)

amdgpu:
● Gralloc Usage Read/Write often swap buffer issues (No text)
● Synchronization issues
● GUI opacity issues/Shadowing between layers

intel:
● EGL_swap_buffer_with_damage causes SIGSEGV MAPERR
● Lack of RGBA_8888 in Primary Planes for older HW Gen
● Cursor subject to ordering problem with HWC2 on Skylake
Status for HW supported with drm Gralloc:

- Intel booting only specific gen (having RGBA support in Primary Planes)
- AMD booting on GCN 2nd gen and later (Atomic required)
- Nvidia booting but has stability issues (GUI restart, GPU lockup)

Status for vGPU supported with drm Gralloc:

- vmwgfx booting with HWC1, but shows black GUI/no icons
- virgl working with HWC1
• AMDGPU drm Gralloc path
• AMDGPU LLVM target support
• LLVM build with new Mesa versions
• SurfaceFlinger RGBA Pixel Format constraint
AMDGPU drm Gralloc path:

- `gralloc_drm_pipe` backend is selected for drivers w/o ad hoc backend
- Updated kernel with `drm/amd/amdgpu` support for SI, CIK, VI, AI
- Updated `mesa`, `libdrm`
- Updated `llvm` with AMDGPU target support
### LLVM versions in android-x86

<table>
<thead>
<tr>
<th>android-x86 branch</th>
<th>LLVM Version</th>
<th>Achievements</th>
</tr>
</thead>
<tbody>
<tr>
<td>lollipop-x86</td>
<td>3.5</td>
<td>AMDGPU target build fix by Paulo Sergio Travaglia (pstiglia)</td>
</tr>
<tr>
<td>marshmallow-x86</td>
<td>3.7</td>
<td>FORCE_AMDGPU cflag to fix function prototypes (maurossi)</td>
</tr>
<tr>
<td>nougat-x86</td>
<td>3.8</td>
<td>AMDGPU target build ported to new llvm version (maurossi)</td>
</tr>
<tr>
<td>oreo-x86 rc1</td>
<td>3.9</td>
<td>AOSP moves to soong build AMDGPU ported (Rob Herring)</td>
</tr>
<tr>
<td>oreo-x86 r1</td>
<td>?</td>
<td>Mesa 18.1 : LLVM 5.0 required but AOSP supports only 3.9</td>
</tr>
</tbody>
</table>

Radeonsi and amdgpu drivers support would have been lost in oreo-x86.
PROBLEM: New LLVM version required, but AOSP supports only LLVM 3.9

- Updated LLVM version does not allow to compile core AOSP projects
- Collaboration started with Qiang Yu (AMD)
- Idea: side build libLLVM50 "for Mesa" separate shared library module
- AOSP does not allow duplicate module, variable: rename them all (!)
- Mesa changes: define libLLVM50 dependency and HAVE_LLVM cflag

Module renames can be reverted to get back "regular Android.mk" build
Procedure to add soong building rules to new LLVM versions:

- Started from AOSP LLVM 3.9 (with soong building rules by Rob Herring)
- Rebase to new LLVM version 5.0
- Fix the build with new LLVM versions
- Have ‘sed’ rename LLVM modules and build variables
- One commit to revert to “go back to normal” with documented changes
- Soong building rules tested on LLVM 5.0, 6.0, 7.0

LLVM 7.0 added to android-x86 repo

LLVM 5.0, 6.0 soong branches: https://github.com/maurossi/llvm
A recurrent problem
• AOSP SurfaceFlinger requires RGBA pixel format (mandatory)
• Workaround 10194508 (Fallback to BGRA in SurfaceFlinger) was removed
• GUI breakage for r300, r600, radeonsi, amdgpu, vmwgfx

How to cope with it
• Adding again equivalent workaround “EGL config fallback to simpler query”
  (but that is not a real solution)

How to solve it
• Adding support for RGBA/RGBX when possible in kernel drm driver
HWC: What happens without RGBA pixel format? (AMDGPU)

```
D [drm:drm_atomic_check_only [drm]] Invalid pixel format AB24 little-endian
I hwc-drm-display-compositor: Commit test pset failed ret=-22
I hwc-drm-display-compositor: Commit test failed, squashing frame for display 0
E hwc-drm-display-compositor: Composite failed for display 0
E hwc-drm-two: Failed to apply the frame composition ret=-22
E HWComposer: presentAndGetReleaseFences: failed for display 0: BadParameter (4)
```

SurfaceFlinger HWcomposer throws error

=> no mode set => Black screen

ABGR Patches submitted to amd-gfx, now in drm-next
(thanks to Alex Deucher for guidance and review)
- Vulkan HAL build integration and initialization
- Vulkan anv CTS test results on Skylake GT2
- Vulkan radv CTS test results on supported GCN
- Vulkan Apps and Games used for testing
- Vulkan HAL outstanding work and next steps
Android makefiles implemented using anv as a reference
Tested with dEQP-VK tests, Benchmarks, Games
Submitted in Mesa 18.2
AOSP build problem: build radv conditionally to radeonsi
Released as part of Mesa 18.2
NOTE: Vulkan HAL anv package, properties and copy files in android-x86

android-x86.org
NOTE: init_hal_vulkan() is added to the sequence of init_hal_*() functions

Set `ro.hardware.vulkan` property
Add *vulkan.radv* new package

Set *ro.hardware.vulkan* property
anv CTS dEQP-VK results

Mesa 18.2.0-devel
(30-Jun-2018)

Very good results

Strange issue with interrupted tests runs
<table>
<thead>
<tr>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>dEQP-VK.api.info.device#extensions</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.api.info.device#properties</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.api.info.instance#extensions</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.cos.highp_compute#scalar</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec2</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec3</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec4</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.sin.highp_compute#scalar</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec2</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec3</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec4</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none_fifo#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none_fifo#reference</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none_mailbox#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up_fifo#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up_fifo#reference</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up_mailbox#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.swapchain#create#image_usage</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.swapchain.simulate_oom#image_usage</td>
<td>fail</td>
</tr>
</tbody>
</table>

Mesa 18.2.0-devel
(30-Jun-2018)

dEQP-VK.wsi.* 10
dEQP-VK.glsl.* 8
dEQP-VK.api.* 3

Total 21

github.com/intel/mesa solves many of these
**radv CTS dEQP-VK results**

<table>
<thead>
<tr>
<th><strong>Suite / Plan</strong></th>
<th>CTS / cts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Suite / Build</strong></td>
<td>8.1_r7 / 4855739</td>
</tr>
<tr>
<td><strong>Host Info</strong></td>
<td>Result@start utente-MS-7576 (Linux - 4.15.0-29-generic)</td>
</tr>
<tr>
<td><strong>Start time / End Time</strong></td>
<td>Sun Jul 22 18:04:12 CEST 2018 / Sun Jul 22 18:18:27 CEST 2018</td>
</tr>
<tr>
<td><strong>Tests Passed</strong></td>
<td>227444</td>
</tr>
<tr>
<td><strong>Tests Failed</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>Modules Done</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Modules Total</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Fingerprint</strong></td>
<td>Android-x86/android_x86_64/x86_64:8.1.0/OPM6.171019.030.E1/maur07141051:userdebug/test-keys</td>
</tr>
<tr>
<td><strong>Security Patch</strong></td>
<td>2018-07-05</td>
</tr>
<tr>
<td><strong>Release (SDK)</strong></td>
<td>8.1.0 (27)</td>
</tr>
<tr>
<td><strong>ABIs</strong></td>
<td>x86_64,x86</td>
</tr>
</tbody>
</table>

---

**Compatibility Program**

**Summary**

<table>
<thead>
<tr>
<th>Module</th>
<th>Passed</th>
<th>Failed</th>
<th>Total Tests</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86 CtsDeqpTestCases</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>true</td>
</tr>
<tr>
<td>x86_64 CtsDeqpTestCases</td>
<td>227444</td>
<td>12</td>
<td>227456</td>
<td>true</td>
</tr>
</tbody>
</table>

---

**Mesa 18.2.0-devel (22-Jul-2018)**

**Very good results**

**CTS Bug reported on multiarch tests re-runs**
### Mesa 18.2.0-devel (22-Jul-2018)

- dEQP-VK.wsi.* 8
- dEQP-VK.api.* 3
- dEQP-VK.spirv.* 1

Total 12

Intel/mesa anv patches could be ported

<table>
<thead>
<tr>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>dEQP-VK.api.info.device#extensions</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.api.info.instance#extensions</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.api.object_management.max_concurrent#device</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.spirv_assembly.instruction.compute.opquantize#propagated_nans</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none.fifo#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none.fifo#reference</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#reference</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up.fifo#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up.fifo#reference</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#incremental_present</td>
<td>fail</td>
</tr>
<tr>
<td>dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#reference</td>
<td>fail</td>
</tr>
</tbody>
</table>
Root cause identified in gbm gralloc Usage Flags for HWC buffer
Patch is now available in gbm gralloc master (robherring github)
<table>
<thead>
<tr>
<th>Android Apps</th>
<th>Android Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DMark - API Overhead</td>
<td>Breakneck</td>
</tr>
<tr>
<td>3DMark - Slingshot Extreme</td>
<td>Dirt Road Transport</td>
</tr>
<tr>
<td>V1 Benchmark Pro</td>
<td>Dream League Soccer</td>
</tr>
<tr>
<td>GFXbench 5.0</td>
<td>Heroes of Incredible Tales</td>
</tr>
<tr>
<td>Google samples tutorials</td>
<td>Olympus Rising</td>
</tr>
<tr>
<td>LunarG examples</td>
<td>Roblox</td>
</tr>
<tr>
<td>Sascha Willems examples</td>
<td>Score! Hero</td>
</tr>
<tr>
<td>VulkanCapsViewer</td>
<td>Sky Force Reloaded</td>
</tr>
<tr>
<td>Dolphin emulator</td>
<td>Toy Zombies Lite</td>
</tr>
<tr>
<td>PPSSPP emulator</td>
<td>Vainglory</td>
</tr>
</tbody>
</table>
Vulkan app issues (test cases)

- GFXbench 5.0 Vulkan tests not executed “OUT_OF_MEMORY” (anv and radv)
- V1 Vulkan Pro display sync issues (radv on Mullins)
- 3DMark “API Overhead” freeze (radv on GCN 2nd gen “GPU fault detected: 146”)
- 3DMark “Slingshot Extreme” orange/black renders (radv GCN 2nd and 4th gen)
Public release of xml manifest, to accelerate fix of pending issues

Complete testing with HWC2

Official iso release when sufficiently stable/usable

Testing of new drivers supporting Atomic Display Framework

Vulkan nouveau, Vulkan virgl, Vulkan SwiftShader (when available)
Questions?

Thank you!