Vulkan-CTS Guide

Khronos Group

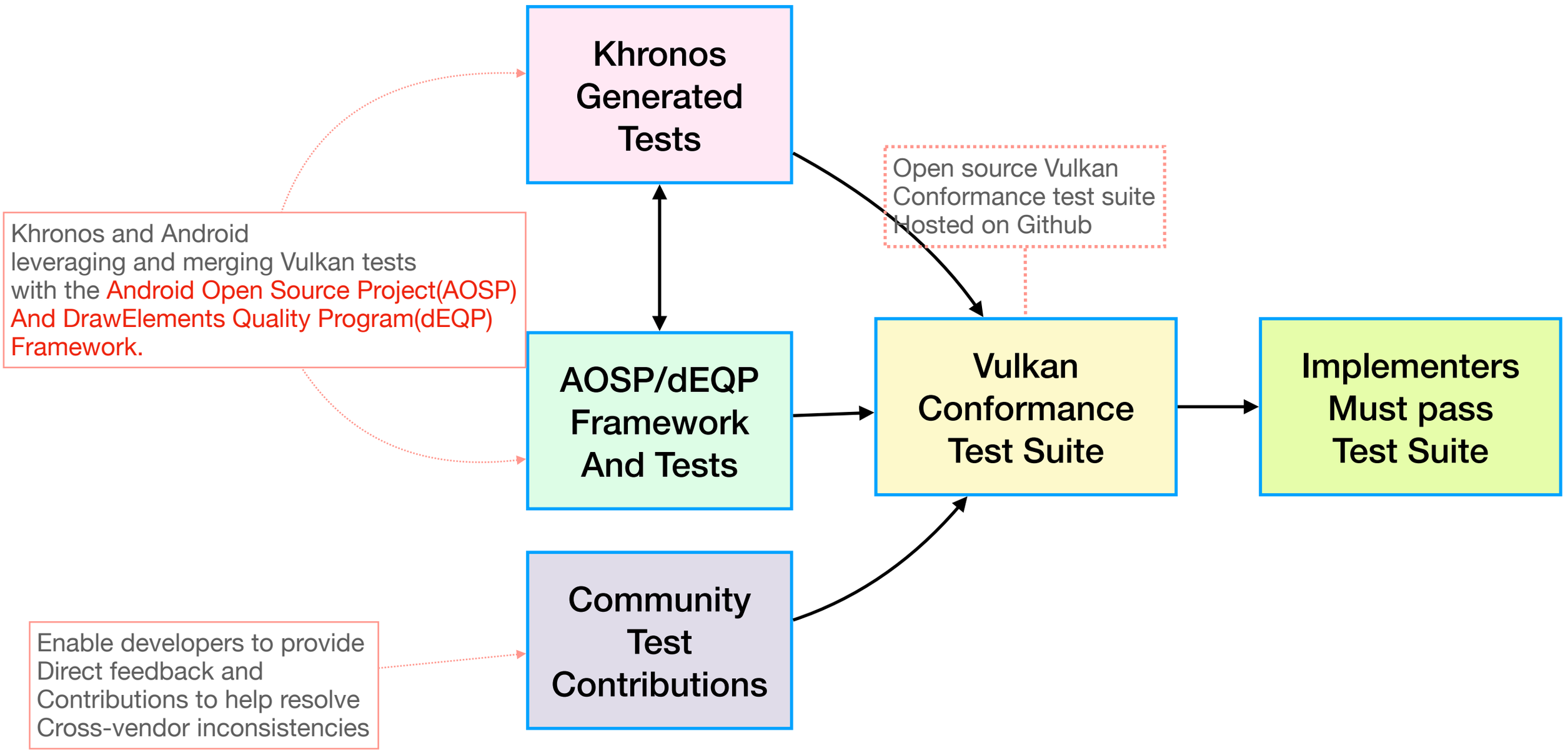
Lina Liu Nov. 22

Vulkan CTS

Vulkan Conformance Tests Suites

1. What is Vulkan CTS?
    1. Vulkan conformance test suite
    2. Is a set of tests used to verify the conformance of an implementation.
    3. The conformant implementation shows that it has successfully passed CTS and it is an valid implementation of Vulkan.
4. Vulkan CTS source code is freely available
    1. Anyone is free to create and add new test
    2. Before add new test, the new test must follow the contributing wiki
        1. https://github.com/KhronosGroup/VK-GL-CTS/wiki/Contributing

Vulkan Open Source Conformance Tests



dEQP

Draw Element Quality Program

1. What's dEQP contained?
    1. GPU testing suite
    2. Contains tests for several graphic APIs, including OpenGL ES, EGL, and Vulkan
    3. Vulkan CTS have been built on dEQP framework
    4. .qpa logs
        1. generated by the conformance tests
        2. Contain embedded PNG images of the results
        3. Can be viewed with scripts/qpa_image_viewer.html
            1. Using the Cherry tool

Running CTS

Cmd line options be used when running CTS

1. Cmd line options be used when running CTS
    1. Running multi test cases at a time:
        1. —deqp-caselist-file=/../../../mustpass/main/vk-default.txt
        2. —deqp-log-images=disable
        3. —deqp-log-shader-sources=disable
2. Run ./deqp-vk —help to get all the cmd line option and it’s meanings

Conformance Submission Package Requirements

1. What's the requirements of the Conformance Submission Package?
    1. Full test logs
        1. TestResult.qpa in the same folder as ./deqp-vk
    2. Run git status git log from vulkancts source directory
    3. Run git cherry-pick for apply bug fixes

Conformance test results

1. Pass
2. NotSupported
3. QualityWarning
4. CompatibilityWarning
5. Waiver

Cherry GUI

Vulkan test module can be used with Cherry

1. Cherry: GUI for test execution and analysis
    1. https://android.googlesource.com/platform/external/cherry

Shader Optimizer

Vulkan CTS can be optionally run with the shader optimizer enabled

1. Experimental feature: Vulkan CTS can run with the shader optimizer enabled.
    1. —deqp-optimization-recipe=<>
2. Experimental feature: Vulkan CTS can run with spir-v optimizer
    1. —deqp-optimize-spirv=enable
        1. Maybe useful in finding new bugs in driver or the optimizer itself.

Shader Cache

Why use Shader Cache?

1. Why use Shader Cache in Vulkan-CTS?
    1. To speed up the running of the CTS
    2. Skipping shader compilation can significantly reduce runtime, especially repeat runs.
    3. Default is shader cache enabled, but truncated at the the start of the CTS run

Shader Cache

How to use Shader Cache

1. —deqp-shadercache=disable
2. —deqp-shadercache-ipc=enable
    1. Enable the use of inter-process communication primitives to allow several instances of CTS to share a single cache file.
    2. All the instances must use the same shader cache filename.
    3. if one instance should crash while holding the cache file lock, the other instances will hang

RenderDoc

(https://renderdoc.org/)

1. What RenderDoc used for?
    1. RenderDoc is graphics debugger
    2. RenderDoc may be used to debug Vulkan Tests
    3. Cmd line option:
        1. —deqp-renderdoc=enable
    2. The cmd line mark each dEQP test case as a separate ‘frame’, just for the purpose of capturing.
    3. The frames are added using RenderDoc ‘In-Application API’, instead of swap chain operations.

Vulkan Safety-Critical Conformance Test suite

Vulkan-CTS framework adapted to Vulkan SC requirements

1. What’s Vulkan SC CTS included?
    1. Contains it own must pass list
    2. Use its own executable module to perform tests:deqp-vksc
    3. Each test in deqp-vksc performed twice
    4. Vulkan SC pipelines may be compiled using offline pipeline compiler delivered by implementation vendor.
    5. Some of the Vulkan SC implementation may not have a possibility to use filesystem, create pipeline caches or log results to file.