Releasing and Testing Free Opensource Graphics Drivers: the case of Mesa3D

Emil Velikov (emil.velikov@collabora.com)
Juan A. Suárez (jasuarez@igalia.com)
with
Pierre-Loup Griffais (pgriffais@valvesoftware.com)

COLLABORA

XDC 2018

igalia

The speakers

- Emil Velikov
  - Software engineer at Collabora
  - Mesa developer since 2011, release manager 2014
- Juan A. Suárez
  - Software engineer at Igalia
  - Mesa developer since 2015, release manager 2017

Agenda

- Introduction to Mesa3D
- Releases
- Historical walk of the release process
- The current process
- Test systems used
- Freedesktop’s GitLab CI
- LunarG’s Mesa3D regression test system

Introduction to Mesa3D

- Started by Brian Paul in 1993 (25 years old!)
- “Framework” to implement graphics drivers supporting different graphics standards: OpenGL/ES, Vulkan, OpenCL, OpenMax, etc
    - Different parts common to all drivers
    - Parts common to many drivers (NIR, Gallivm, etc)
    - Drivers targeting many vendors
        - Official drivers from Intel
        - Unofficial drivers from AMD and NVidia
        - ARM drivers - Qualcomm, Broadcom, Vivante
        - Two virtual drivers - VMware and VirGL
        - Four software drivers

Releases

- Feature releases
    - Big releases with new features
    - 4 in a year (one per quarter, more or less): Mesa YEAR.X.0, with X={0..3}
    - Started as branch point in master
    - Apply patches to stabilize and fix bugs
    - Create a RC release per week, around 4 weeks until everything is fine
    - Create final release from last RC
- Bugfix releases
    - No new features, only fixes
    - One release every two weeks
    - Last release after first feature release

Project origin and early process

- Mesa 1.0 beta in February 1995
- Releasing handled by Brian Paul
- In early development stage
- No documentation of the process
- No distinct feature/bugfix releases

- Mesa feature/bugfix releases since 3.2
- Limited bugfix releases, 2-6 months between
- Noticeable improvements circa 6.4 - 7.0
- Conducts 2-3 stable releases, 1-4 months apart

A more formal process

- Intel's Ian Romanick step after Brian
- Mesa 7.6, circa 2009
- Improves quality and frequency of bugfix releases - 2-3, monthly
- Introduces a tag for nomination:
  NOTE: this is a candidate for back-porting to the X.Y stable branch.

A more formal process (2)

- Intel's Carl Worth starts helping with bugfix releases
- Mesa 9.1, circa 2013
- Handles 6-9 bugfix releases, out every fortnight
- Introduces *CC: mesa-stable@* deprecates earlier *NOTE*
- Formulates the acceptance criteria
- Documents the process, shortly before handing it over

Document everything

- Emil Velikov steps in, after Ian and Carl
- Mesa 10.3, back in 2014
- Makes the releasing process MT
- Build test all* of Mesa - OSMesa, Nine, OpenCL...
- Build test on more platforms
    - Linux: w/ and w/o libdrm (locally), Travis
    - Windows: MinGW-w64 (locally), AppVeyor
- Refactored and doubled the releasing documentation
- Improved existing nominations scripts
- Introduces Fixes tag

More than one release manager

- Andres and Juan from Igalia helping out since 17.0
- Initially helping out with bugfix release
- Minor misunderstandings who's doing which release
- Added a release table - preliminary dates, release managers
- Further tweaks to the scripts
- Working on Gitlab CI

Fresh blood

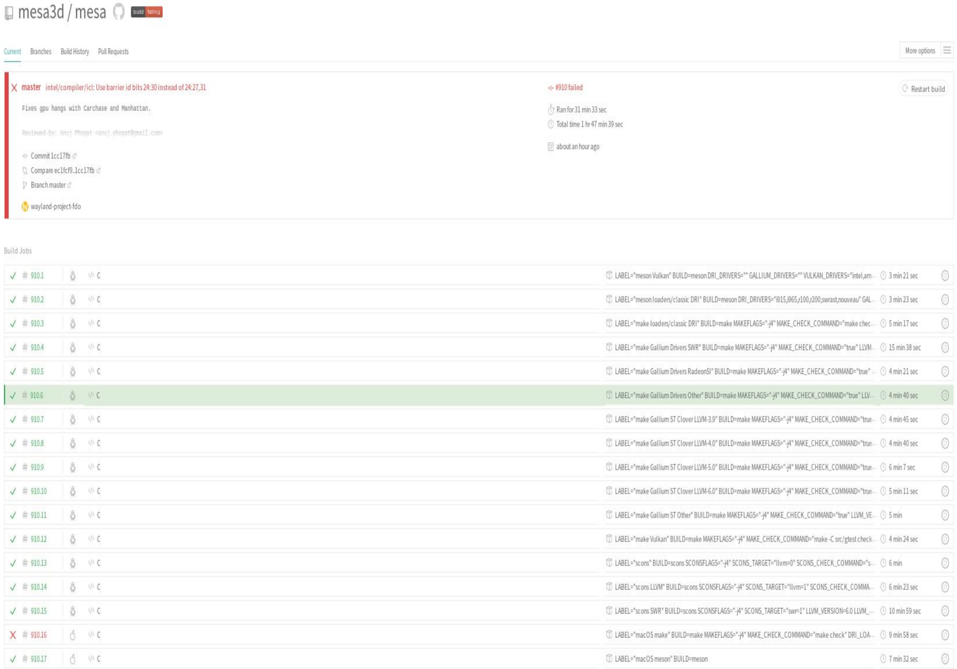
- Dylan from Intel, helping out since 18.1
- Resident Python expert, helping with Mesa and Piglit python code
- Direct access to the Intel CI, more on that later

CC vs Fixes

- *CC: mesa-stable@*
  - simplifies managers' job, and allows later nominations
  - separates important fixes from the huge volume at *mesa-dev@*
  - use when the offending commit is none/unknown
- *Fixes*
  - consistently annotates the origin of the problem
  - shows maintainer for which stable branches patch is applicable
  - while, developers don't need to bother knowing
- Will my patch get dropped silently?
  - **No**, not even when the patch is self-rejected
  - Release managers makes their best effort to apply the patches
  - For patches which are not merged, the manager will inform author/nominator

Is the release buildable?

- Several build tools
  - autotools, scons and meson
- Drivers depends on LLVM
  - Different versions
  - Different APIs
- Detect as earlier as possible
- Not only the release, but also master
  - Automatic system: GitHub + Travis + AppVeyor



mesa3d / mesa

✗ master intel/compiler/icl: Use barrier id bits 24:30 instead of 24:27,31

```
Fixes gpu hangs with Carchase and Manhattan.

Reviewed-by: Anuj Phogat <anuj.phogat@gmail.com>
```

Commit 1cc17fb
Compare ec1fcf9..1cc17fb
Branch master

wayland-project-fdo

#910 failed
Ran for 31 min 33 sec
Total time 1 hr 47 min 39 sec
about an hour ago

Build Jobs

✓	# 910.1	C	LABEL="meson Vulkan" BUILD=meson DRI_DRIVERS="" GALLIUM_DRIVERS="" VULKAN_DRIVERS="intel,am...	3 min 21 sec
✓	# 910.2	C	LABEL="meson loaders/classic DRI" BUILD=meson DRI_DRIVERS="i915,i965,r100,r200,swrast,nouveau" GAL...	3 min 23 sec
✓	# 910.3	C	LABEL="make loaders/classic DRI" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="make chec...	5 min 17 sec
✓	# 910.4	C	LABEL="make Gallium Drivers SWR" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true" LLVM...	15 min 38 sec
✓	# 910.5	C	LABEL="make Gallium Drivers RadeonSI" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true" ...	4 min 21 sec
✓	# 910.6	C	LABEL="make Gallium Drivers Other" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true" LLV...	4 min 40 sec
✓	# 910.7	C	LABEL="make Gallium ST Clover LLVM-3.9" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true...	4 min 45 sec
✓	# 910.8	C	LABEL="make Gallium ST Clover LLVM-4.0" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true...	4 min 40 sec
✓	# 910.9	C	LABEL="make Gallium ST Clover LLVM-5.0" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true...	6 min 7 sec
✓	# 910.10	C	LABEL="make Gallium ST Clover LLVM-6.0" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true...	5 min 11 sec
✓	# 910.11	C	LABEL="make Gallium ST Other" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="true" LLVM_VE...	5 min
✓	# 910.12	C	LABEL="make Vulkan" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="make -C src/gtest check...	4 min 24 sec
✓	# 910.13	C	LABEL="scons" BUILD=scons SCONSFLAGS="-j4" SCONS_TARGET="llvm=0" SCONS_CHECK_COMMAND="s...	6 min
✓	# 910.14	C	LABEL="scons LLVM" BUILD=scons SCONSFLAGS="-j4" SCONS_TARGET="llvm=1" SCONS_CHECK_COMMA...	6 min 23 sec
✓	# 910.15	C	LABEL="scons SWR" BUILD=scons SCONSFLAGS="-j4" SCONS_TARGET="swr=1" LLVM_VERSION=6.0 LLVM_...	10 min 59 sec
✗	# 910.16	C	LABEL="macOS make" BUILD=make MAKEFLAGS="-j4" MAKE_CHECK_COMMAND="make check" DRI_LOA...	9 min 58 sec
✓	# 910.17	C	LABEL="macOS meson" BUILD=meson	7 min 32 sec

Is the release working?

- Check if it builds is not enough
- Check it actually works => testing
  - Manual testing: test suites, games, 3D apps, etc
  - Automated testing
- Different types of tests
  - Unit testing
  - Functional testing

Is the release working?

- Check if it builds is not enough
- Check it actually works => testing
  - Manual testing: test suites, games, 3D apps, etc
  - Automated testing
- Different types of tests
  - **Unit testing**
  - Functional testing

Does the release has bugs?

- Intel CI
    - Very powerful and useful CI system
    - Used frequently also by developers
    - Basic tool for release managers
    - Required to success before making the release
        - Running this test process takes lot of time
        - For any late (critical) patches the testing has to be redone => almost delay
        - Note: it means that (non-critical) patches arriving during this process, will be delayed for next release (Nominated patches)
- Thoroughly explained in next talk
    - *Mark Janes & Clayton Craft - Mesa Continuous Integration at Intel*

Improving our testing

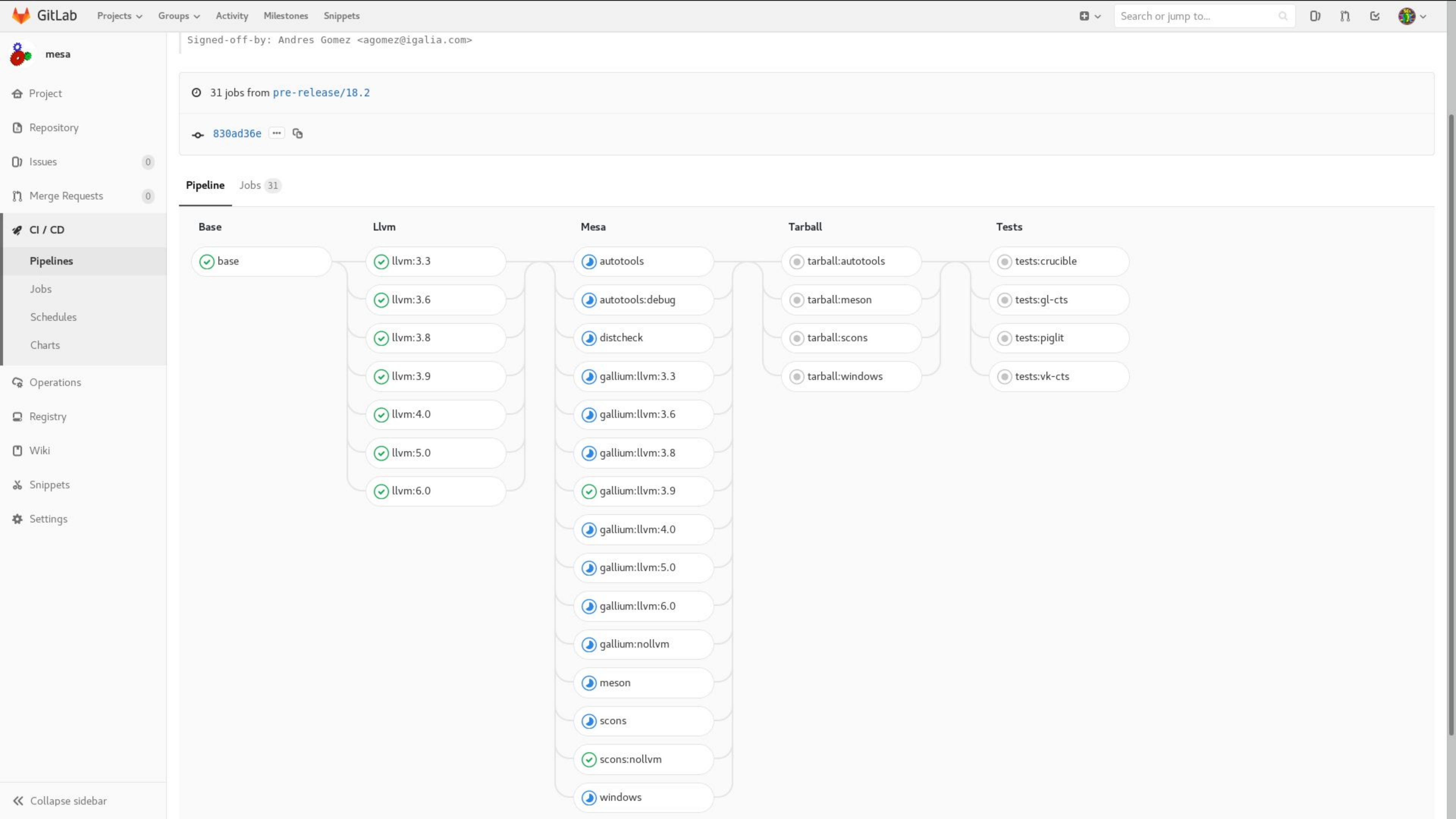
- So far, main repository + GitHub + Travis CI + AppVeyor
- Now, we have GitLab in Freedesktop
  - Check *Daniel Stone & Keith Packard - freedesktop.org update* talk
  - It provides repositories
  - It provides a Continuous Integration system
  - It allows your own runners
  - Many other features
- Igalia using GitLab[.com] during several releases as our own CI
  - Used only when preparing releases
  - Detect as much as possible regressions in earlier stages
  - Used as previous step before using Intel CI

GitLab CI

- Premise: build once, test everywhere
  - Reduce the whole build + testing time
  - Try to use the same configuration in all tests
  - Allow to use not-so-powerful hosts for testing
- Need an easy way to store the build artifacts and re-use them in all the testing hosts
  - Containers
  - GitLab Registry
  - Easy to (re-)generate locally

GitLab CI: building

- Create several images using different build tools and different LLVM versions
    - As in Travis, ensures that Mesa3D can be built
    - Use Rocker to build docker images: templates, mounts on build time, single executable
- Only keep one image
    - This contains all the drivers we want to test
- Avoid re-building and installing all the dependencies required
    - Create a base image with the dependencies plus different images with different LLVM versions
    - Only re-build them if there are new dependencies or changes
    - Force a rebuild once per week to ensure we always get the last updates from the Linux distribution



```
Signed-off-by: Andres Gomez <agomez@igalia.com>
```

31 jobs from `pre-release/18.2`

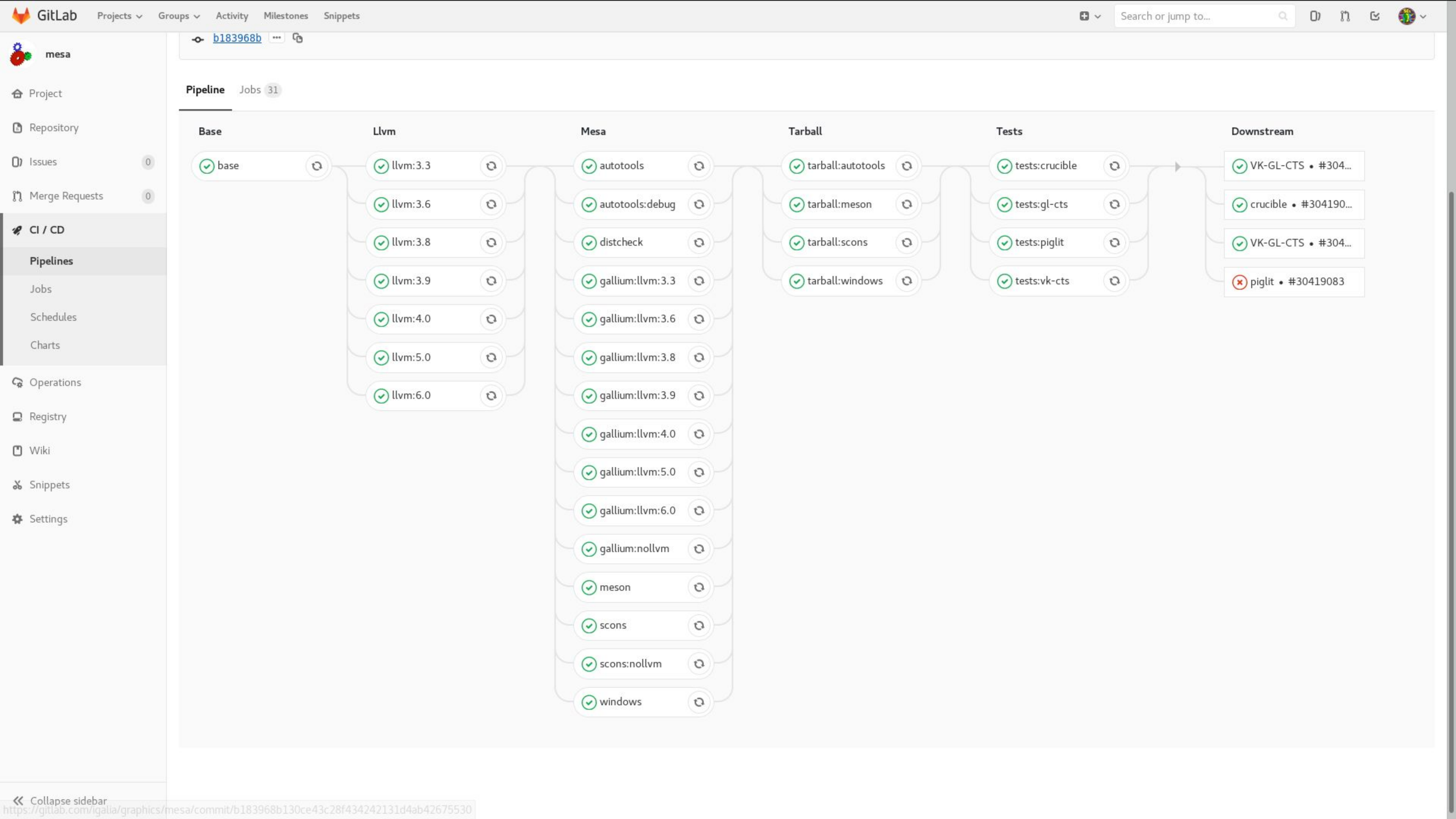
`830ad36e`

Pipeline Jobs 31

Base	Llvm	Mesa	Tarball	Tests
✓ base	✓ llvm:3.3	◐ autotools	○ tarball:autotools	○ tests:crucible
	✓ llvm:3.6	◐ autotools:debug	○ tarball:meson	○ tests:gl-cts
	✓ llvm:3.8	◐ distcheck	○ tarball:scons	○ tests:piglit
	✓ llvm:3.9	◐ gallium:llvm:3.3	○ tarball:windows	○ tests:vk-cts
	✓ llvm:4.0	◐ gallium:llvm:3.6		
	✓ llvm:5.0	◐ gallium:llvm:3.8		
	✓ llvm:6.0	✓ gallium:llvm:3.9		
		◐ gallium:llvm:4.0		
		◐ gallium:llvm:5.0		
		◐ gallium:llvm:6.0		
		◐ gallium:nollvm		
		◐ meson		
		◐ scons		
		✓ scons:nollvm		
		◐ windows		

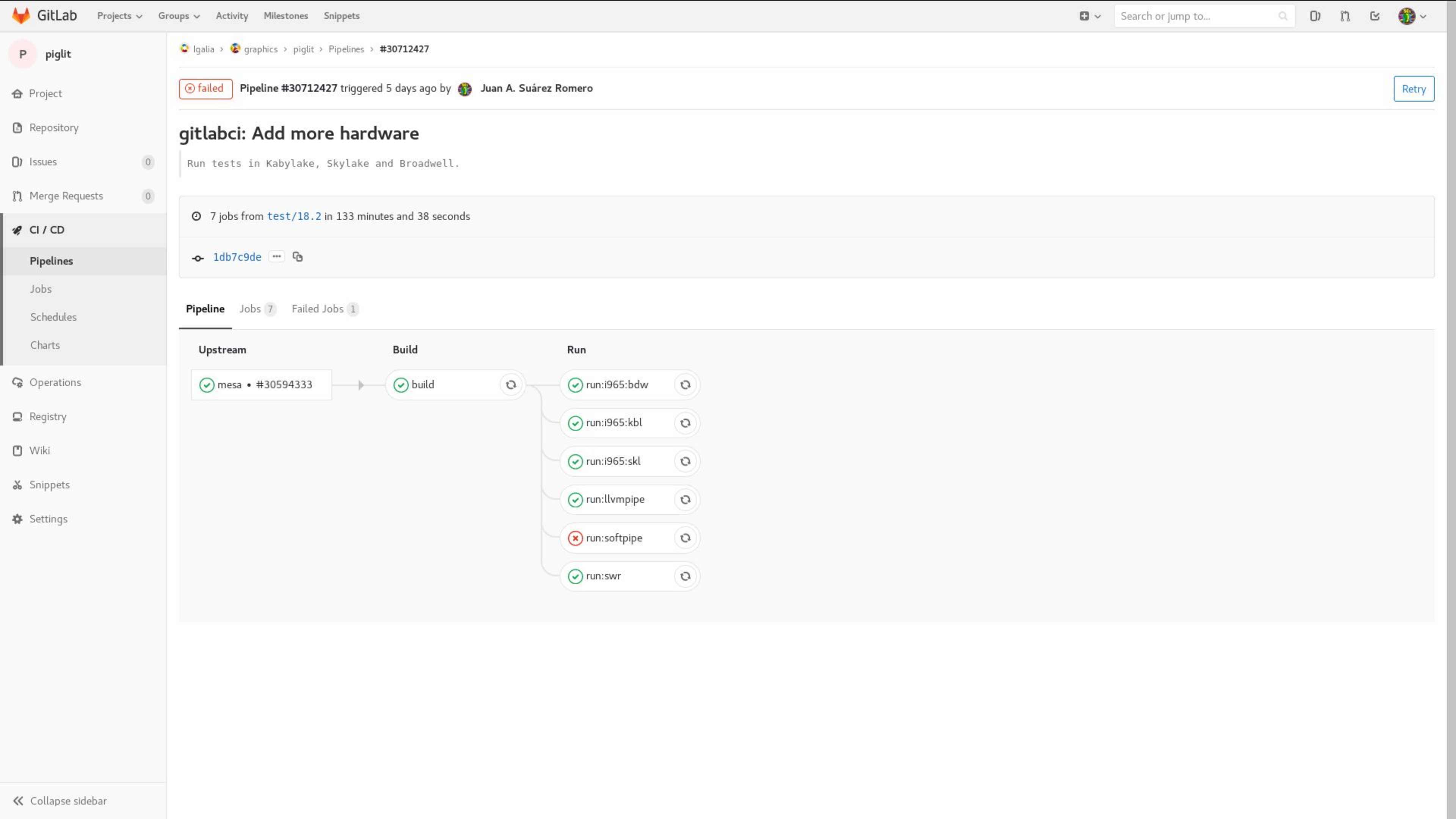
GitLab CI: testing

- Need real hardware with graphics cards
- GitLab allows to provide your own runners
  - Different executors: SSH, Docker, VirtualBox, etc
  - Our case: Docker + mounting graphics device
  - Use tags to match testing jobs with specific hardware
- Trigger pipeline execution in other projects
  - Main build in Mesa3D repository
  - Triggers test building and running in other repositories
    - Piglit
    - Vulkan/OpenGL CTS
    - Crucible
  - Allows to browse between projects



Base	Llvm	Mesa	Tarball	Tests	Downstream
✓ base	✓ llvm:3.3	✓ autotools	✓ tarball:autotools	✓ tests:crucible	✓ VK-GL-CTS • #304...
	✓ llvm:3.6	✓ autotools:debug	✓ tarball:meson	✓ tests:gl-cts	✓ crucible • #304190...
	✓ llvm:3.8	✓ distcheck	✓ tarball:scons	✓ tests:piglit	✓ VK-GL-CTS • #304...
	✓ llvm:3.9	✓ gallium:llvm:3.3	✓ tarball:windows	✓ tests:vk-cts	✗ piglit • #30419083
	✓ llvm:4.0	✓ gallium:llvm:3.6			
	✓ llvm:5.0	✓ gallium:llvm:3.8			
	✓ llvm:6.0	✓ gallium:llvm:3.9			
		✓ gallium:llvm:4.0			
		✓ gallium:llvm:5.0			
		✓ gallium:llvm:6.0			
		✓ gallium:nollvm			
		✓ meson			
		✓ scons			
		✓ scons:nollvm			
		✓ windows			

https://gitlab.com/igalia/graphics/mesa/commit/b183968b130ce43c28f434242131d4ab42675530



failed **Pipeline #30712427** triggered 5 days ago by **Juan A. Suárez Romero**

gitlabci: Add more hardware

```
Run tests in Kabylake, Skylake and Broadwell.
```

7 jobs from `test/18.2` in 133 minutes and 38 seconds

`1db7c9de`

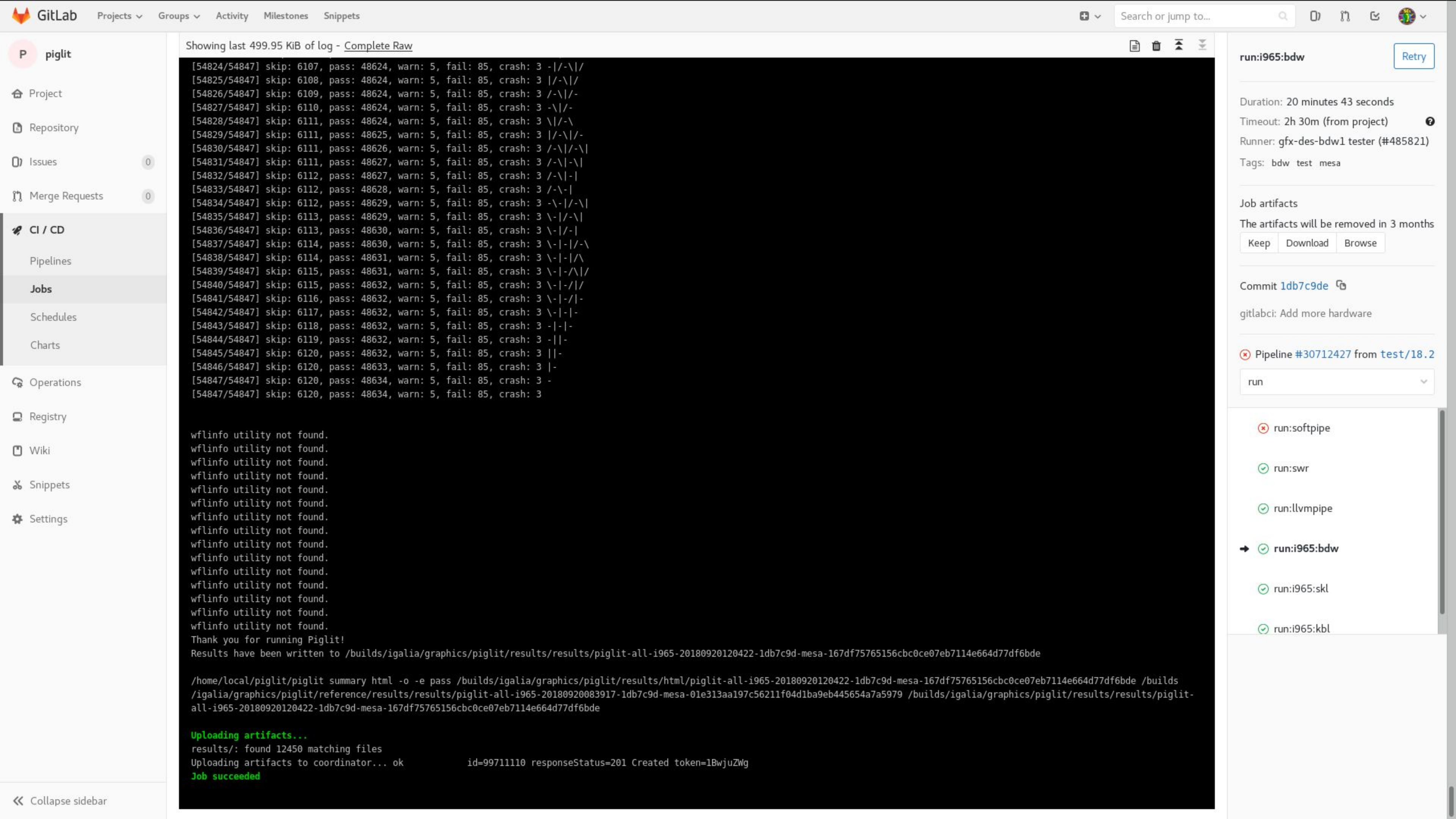
Upstream
- ✓ mesa • #30594333

Build
- ✓ build

Run
- ✓ run:i965:bdw
- ✓ run:i965:kbl
- ✓ run:i965:skl
- ✓ run:llvmpipe
- ✗ run:softpipe
- ✓ run:swr

GitLab CI: testing

- Shows the test results in HTML (use piglit to run the tests)
- Exported as an GitLab's artifact
- Use test results from last release as reference
  - Run a simplified version for releases; results are the reference ones
  - Detect regressions in the pre-release



Showing last 499.95 KiB of log - Complete Raw

```
[54824/54847] skip: 6107, pass: 48624, warn: 5, fail: 85, crash: 3 -|/-\|/
[54825/54847] skip: 6108, pass: 48624, warn: 5, fail: 85, crash: 3 |/-\|/
[54826/54847] skip: 6109, pass: 48624, warn: 5, fail: 85, crash: 3 /-\|/-
[54827/54847] skip: 6110, pass: 48624, warn: 5, fail: 85, crash: 3 -\|/-
[54828/54847] skip: 6111, pass: 48624, warn: 5, fail: 85, crash: 3 \|/-\
[54829/54847] skip: 6111, pass: 48625, warn: 5, fail: 85, crash: 3 |/-\|/-
[54830/54847] skip: 6111, pass: 48626, warn: 5, fail: 85, crash: 3 /-\|/-\|
[54831/54847] skip: 6111, pass: 48627, warn: 5, fail: 85, crash: 3 /-\|-\|
[54832/54847] skip: 6112, pass: 48627, warn: 5, fail: 85, crash: 3 /-\|-|
[54833/54847] skip: 6112, pass: 48628, warn: 5, fail: 85, crash: 3 /-\-|
[54834/54847] skip: 6112, pass: 48629, warn: 5, fail: 85, crash: 3 -\-|/-\|
[54835/54847] skip: 6113, pass: 48629, warn: 5, fail: 85, crash: 3 \-|/-\|
[54836/54847] skip: 6113, pass: 48630, warn: 5, fail: 85, crash: 3 \-|/-|
[54837/54847] skip: 6114, pass: 48630, warn: 5, fail: 85, crash: 3 \-|-|/-\
[54838/54847] skip: 6114, pass: 48631, warn: 5, fail: 85, crash: 3 \-|-|/\
[54839/54847] skip: 6115, pass: 48631, warn: 5, fail: 85, crash: 3 \-|-/\|/
[54840/54847] skip: 6115, pass: 48632, warn: 5, fail: 85, crash: 3 \-|-/|/
[54841/54847] skip: 6116, pass: 48632, warn: 5, fail: 85, crash: 3 \-|-/|-
[54842/54847] skip: 6117, pass: 48632, warn: 5, fail: 85, crash: 3 \-|-|-
[54843/54847] skip: 6118, pass: 48632, warn: 5, fail: 85, crash: 3 -|-|-
[54844/54847] skip: 6119, pass: 48632, warn: 5, fail: 85, crash: 3 -||-
[54845/54847] skip: 6120, pass: 48632, warn: 5, fail: 85, crash: 3 ||-
[54846/54847] skip: 6120, pass: 48633, warn: 5, fail: 85, crash: 3 |-
[54847/54847] skip: 6120, pass: 48634, warn: 5, fail: 85, crash: 3 -
[54847/54847] skip: 6120, pass: 48634, warn: 5, fail: 85, crash: 3


wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
wflinfo utility not found.
Thank you for running Piglit!
Results have been written to /builds/igalia/graphics/piglit/results/results/piglit-all-i965-20180920120422-1db7c9d-mesa-167df75765156cbc0ce07eb7114e664d77df6bde

/home/local/piglit/piglit summary html -o -e pass /builds/igalia/graphics/piglit/results/html/piglit-all-i965-20180920120422-1db7c9d-mesa-167df75765156cbc0ce07eb7114e664d77df6bde /builds
/igalia/graphics/piglit/reference/results/results/piglit-all-i965-20180920083917-1db7c9d-mesa-01e313aa197c56211f04d1ba9eb445654a7a5979 /builds/igalia/graphics/piglit/results/results/piglit-
all-i965-20180920120422-1db7c9d-mesa-167df75765156cbc0ce07eb7114e664d77df6bde

Uploading artifacts...
results/: found 12450 matching files
Uploading artifacts to coordinator... ok            id=99711110 responseStatus=201 Created token=1BwjuZWg
Job succeeded
```

run:i965:bdw

Duration: 20 minutes 43 seconds

Timeout: 2h 30m (from project)

Runner: gfx-des-bdw1 tester (#485821)

Tags: bdw test mesa

Job artifacts

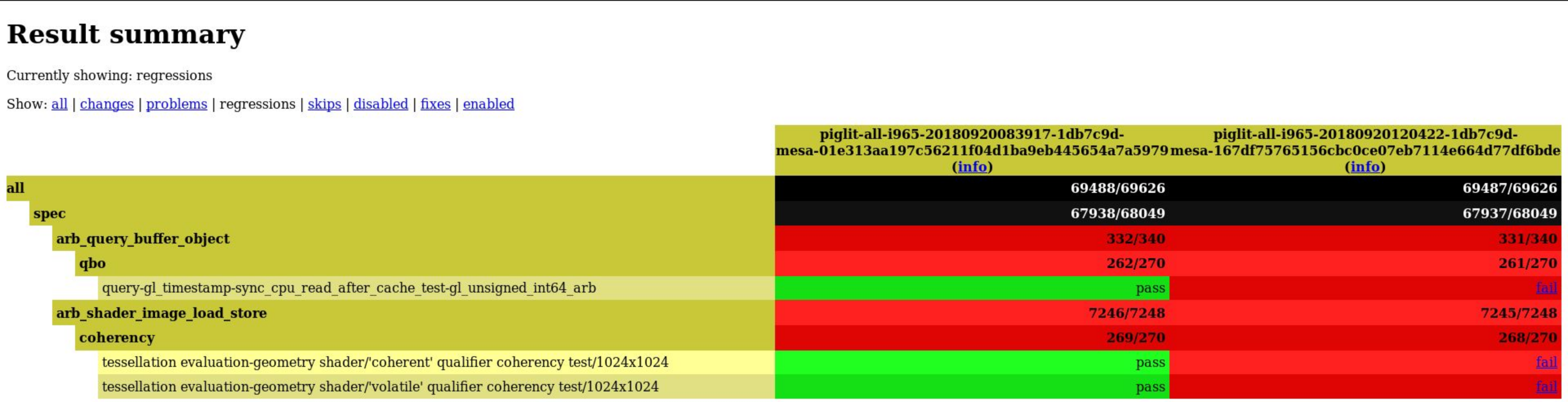
The artifacts will be removed in 3 months

Commit 1db7c9de

gitlabci: Add more hardware

Pipeline #30712427 from test/18.2

- run:softpipe
- run:swr
- run:llvmpipe
- run:i965:bdw
- run:i965:skl
- run:i965:kbl



Result summary

Currently showing: regressions

Show: all | changes | problems | regressions | skips | disabled | fixes | enabled

	piglit-all-i965-20180920083917-1db7c9d-mesa-01e313aa197c56211f04d1ba9eb445654a7a5979 (info)	piglit-all-i965-20180920120422-1db7c9d-mesa-167df75765156cbc0ce07eb7114e664d77df6bde (info)
all	**69488/69626**	**69487/69626**
spec	**67938/68049**	**67937/68049**
arb_query_buffer_object	**332/340**	**331/340**
qbo	**262/270**	**261/270**
query-gl_timestamp-sync_cpu_read_after_cache_test-gl_unsigned_int64_arb	pass	fail
arb_shader_image_load_store	**7246/7248**	**7245/7248**
coherency	**269/270**	**268/270**
tessellation evaluation-geometry shader/'coherent' qualifier coherency test/1024x1024	pass	fail
tessellation evaluation-geometry shader/'volatile' qualifier coherency test/1024x1024	pass	fail

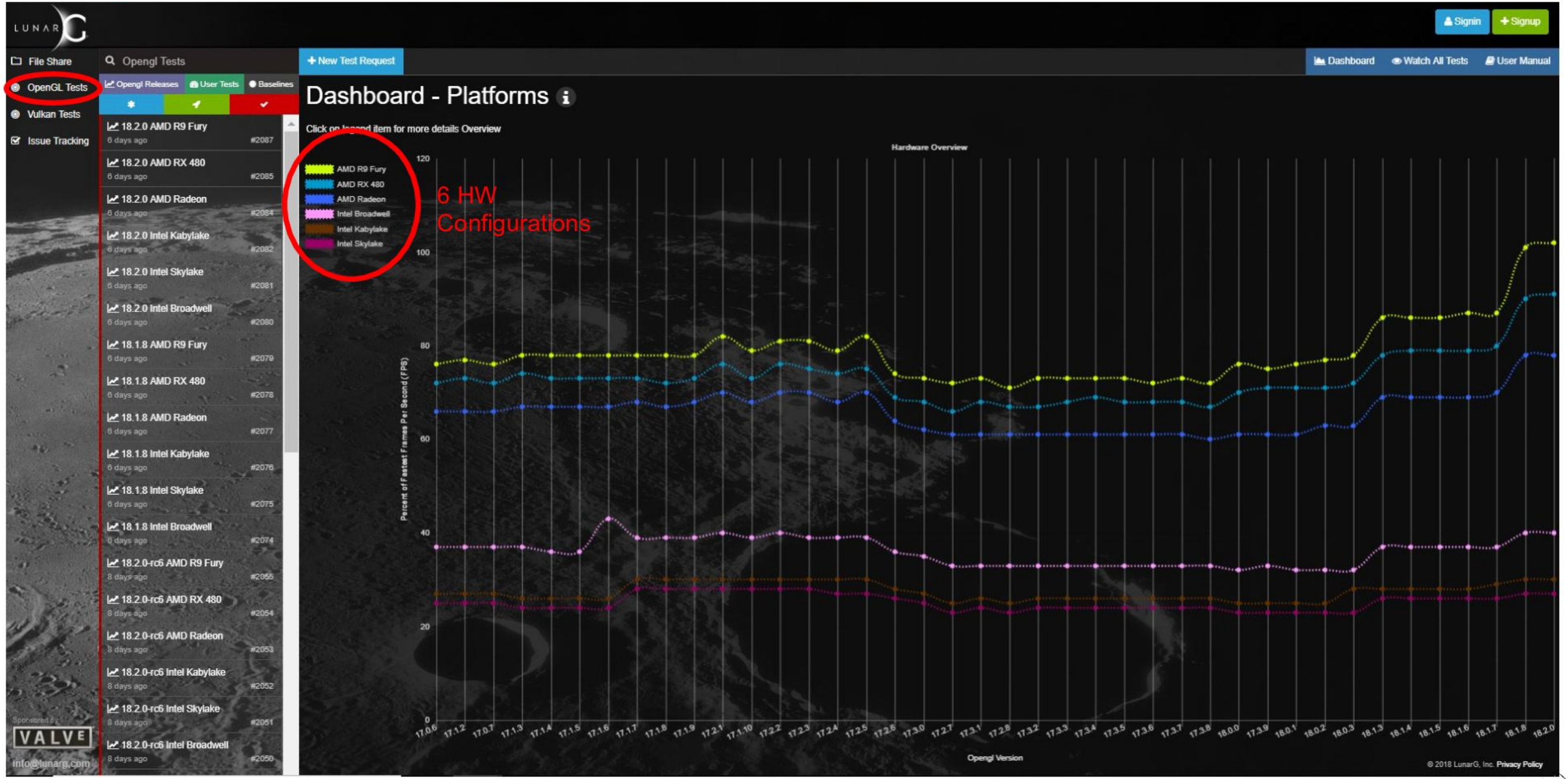
Is the release working?

- Check if it builds is not enough
- Check it actually works => testing
  - Manual testing: test suites, games, 3D apps, etc
  - Automated testing
- Different types of tests
  - Unit testing
  - **Functional testing**

LunarG's Mesa Driver Regression Testing

- Sponsored by Valve
- Testing OpenGL and Vulkan Mesa drivers
- Objectives
  - Regression detection from one Mesa release to the next (open to public)
  - Service to Mesa graphics driver developers (creates account to test personal branches)
  - Ongoing testing of Mesa releases to build a history of results and ongoing release quality monitoring
  - NOT a performance benchmarking test suite
- Methodology
  - Capture traces from Steam Linux games
  - Replay traces on each Mesa release looking for image and performance regressions

OpenGL Driver Regression Testing (https://share.lunarg.com/opengl/home)



18.2.0 AMD R9 Fury
6 days ago #2087

18.2.0 AMD RX 480
6 days ago #2085

18.2.0 AMD Radeon
6 days ago #2084

18.2.0 Intel Kabylake
6 days ago #2082

18.2.0 Intel Skylake
6 days ago #2081

18.2.0 Intel Broadwell
6 days ago #2080

18.1.8 AMD R9 Fury
6 days ago #2079

18.1.8 AMD RX 480
6 days ago #2078

18.1.8 AMD Radeon
6 days ago #2077

18.1.8 Intel Kabylake
6 days ago #2076

18.1.8 Intel Skylake
6 days ago #2075

18.1.8 Intel Broadwell
6 days ago #2074

18.2.0-rc6 AMD R9 Fury
8 days ago #2055

18.2.0-rc6 AMD RX 480
8 days ago #2054

18.2.0-rc6 AMD Radeon
8 days ago #2053

18.2.0-rc6 Intel Kabylake
8 days ago #2052

18.2.0-rc6 Intel Skylake
8 days ago #2051

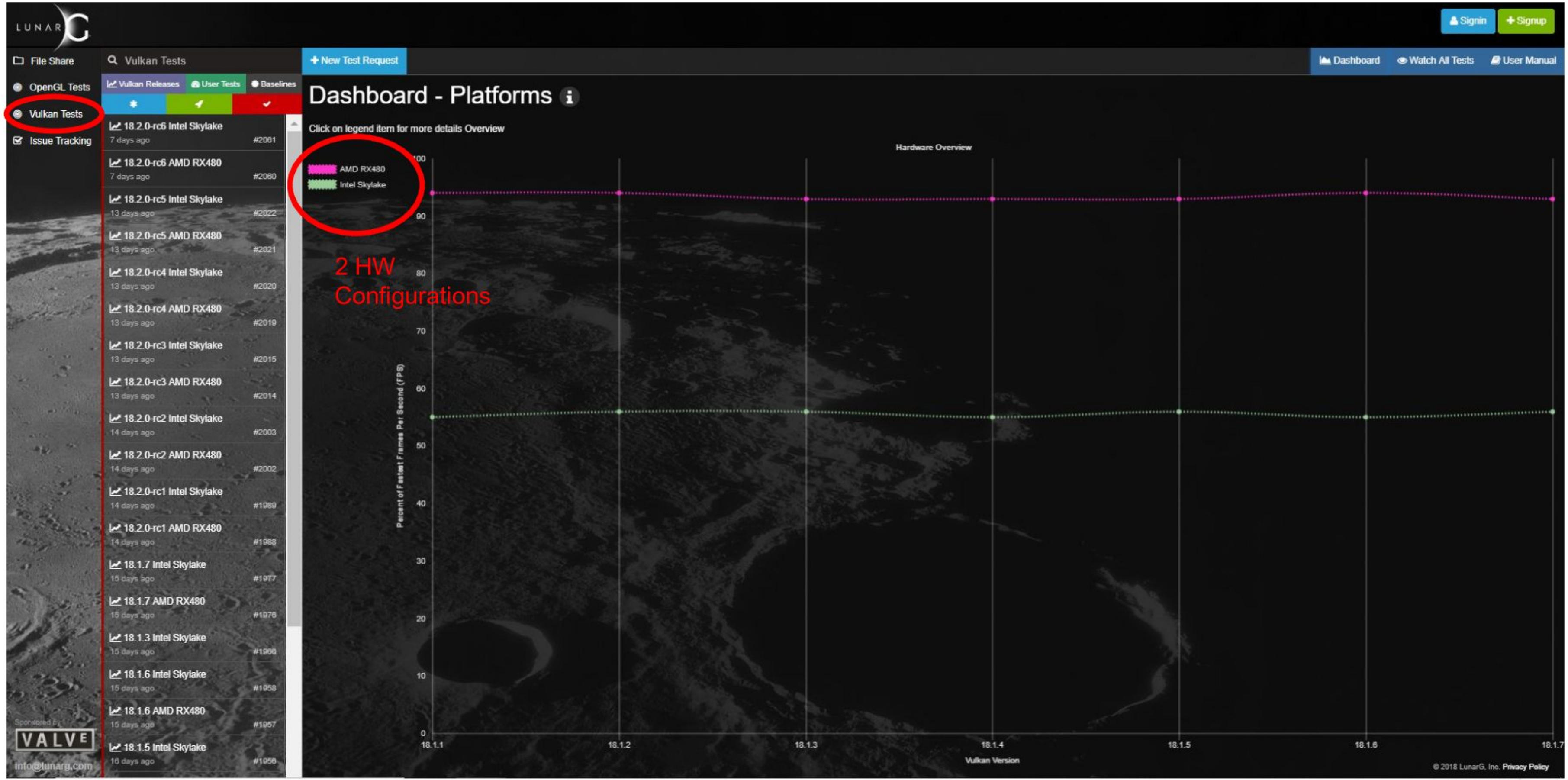
18.2.0-rc6 Intel Broadwell
8 days ago #2050

Dashboard - Platforms

Click on legend item for more details Overview

6 HW Configurations

Vulkan Driver Regression Testing (https://share.lunarg.com/vulkan/home)

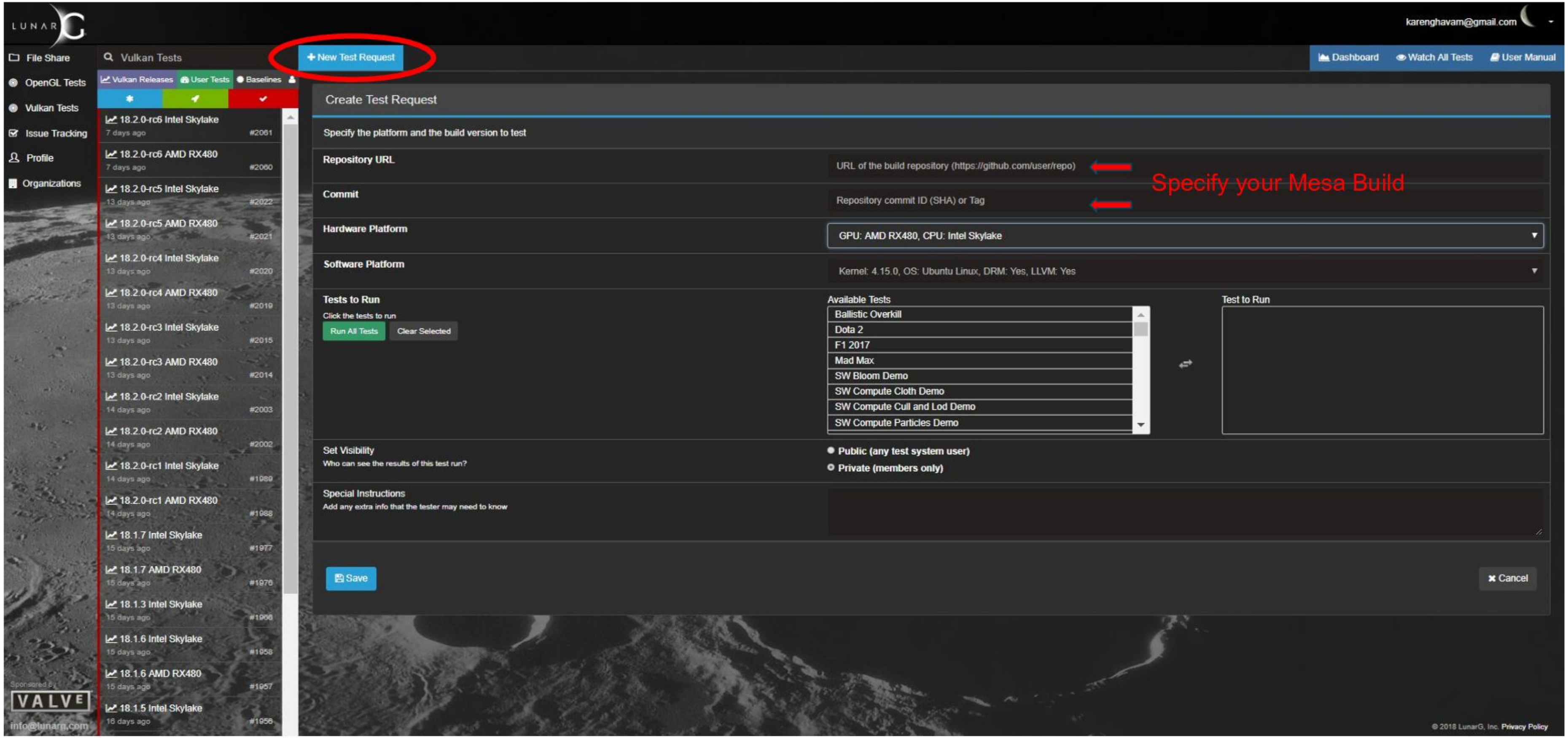


18.2.0-rc6 Intel Skylake — 7 days ago — #2061
18.2.0-rc6 AMD RX480 — 7 days ago — #2060
18.2.0-rc5 Intel Skylake — 13 days ago — #2022
18.2.0-rc5 AMD RX480 — 13 days ago — #2021
18.2.0-rc4 Intel Skylake — 13 days ago — #2020
18.2.0-rc4 AMD RX480 — 13 days ago — #2019
18.2.0-rc3 Intel Skylake — 13 days ago — #2015
18.2.0-rc3 AMD RX480 — 13 days ago — #2014
18.2.0-rc2 Intel Skylake — 14 days ago — #2003
18.2.0-rc2 AMD RX480 — 14 days ago — #2002
18.2.0-rc1 Intel Skylake — 14 days ago — #1989
18.2.0-rc1 AMD RX480 — 14 days ago — #1988
18.1.7 Intel Skylake — 15 days ago — #1977
18.1.7 AMD RX480 — 15 days ago — #1976
18.1.3 Intel Skylake — 15 days ago — #1966
18.1.6 Intel Skylake — 15 days ago — #1958
18.1.6 AMD RX480 — 15 days ago — #1957
18.1.5 Intel Skylake — 16 days ago — #1956

Click on legend item for more details Overview

2 HW Configurations

Request a Test Run (https://share.lunarg.com/vulkan/create)



- 18.2.0-rc6 Intel Skylake — 7 days ago — #2061
- 18.2.0-rc6 AMD RX480 — 7 days ago — #2060
- 18.2.0-rc5 Intel Skylake — 13 days ago — #2022
- 18.2.0-rc5 AMD RX480 — 13 days ago — #2021
- 18.2.0-rc4 Intel Skylake — 13 days ago — #2020
- 18.2.0-rc4 AMD RX480 — 13 days ago — #2019
- 18.2.0-rc3 Intel Skylake — 13 days ago — #2015
- 18.2.0-rc3 AMD RX480 — 13 days ago — #2014
- 18.2.0-rc2 Intel Skylake — 14 days ago — #2003
- 18.2.0-rc2 AMD RX480 — 14 days ago — #2002
- 18.2.0-rc1 Intel Skylake — 14 days ago — #1989
- 18.2.0-rc1 AMD RX480 — 14 days ago — #1988
- 18.1.7 Intel Skylake — 15 days ago — #1977
- 18.1.7 AMD RX480 — 15 days ago — #1976
- 18.1.3 Intel Skylake — 15 days ago — #1966
- 18.1.6 Intel Skylake — 15 days ago — #1958
- 18.1.6 AMD RX480 — 15 days ago — #1957
- 18.1.5 Intel Skylake — 16 days ago — #1956

Create Test Request

Specify the platform and the build version to test

Repository URL — URL of the build repository (https://github.com/user/repo)

Commit — Repository commit ID (SHA) or Tag

Specify your Mesa Build

Hardware Platform — GPU: AMD RX480, CPU: Intel Skylake

Software Platform — Kernel: 4.15.0, OS: Ubuntu Linux, DRM: Yes, LLVM: Yes

Tests to Run

Click the tests to run

Available Tests:

- Ballistic Overkill
- Dota 2
- F1 2017
- Mad Max
- SW Bloom Demo
- SW Compute Cloth Demo
- SW Compute Cull and Lod Demo
- SW Compute Particles Demo

Test to Run

Set Visibility

Who can see the results of this test run?

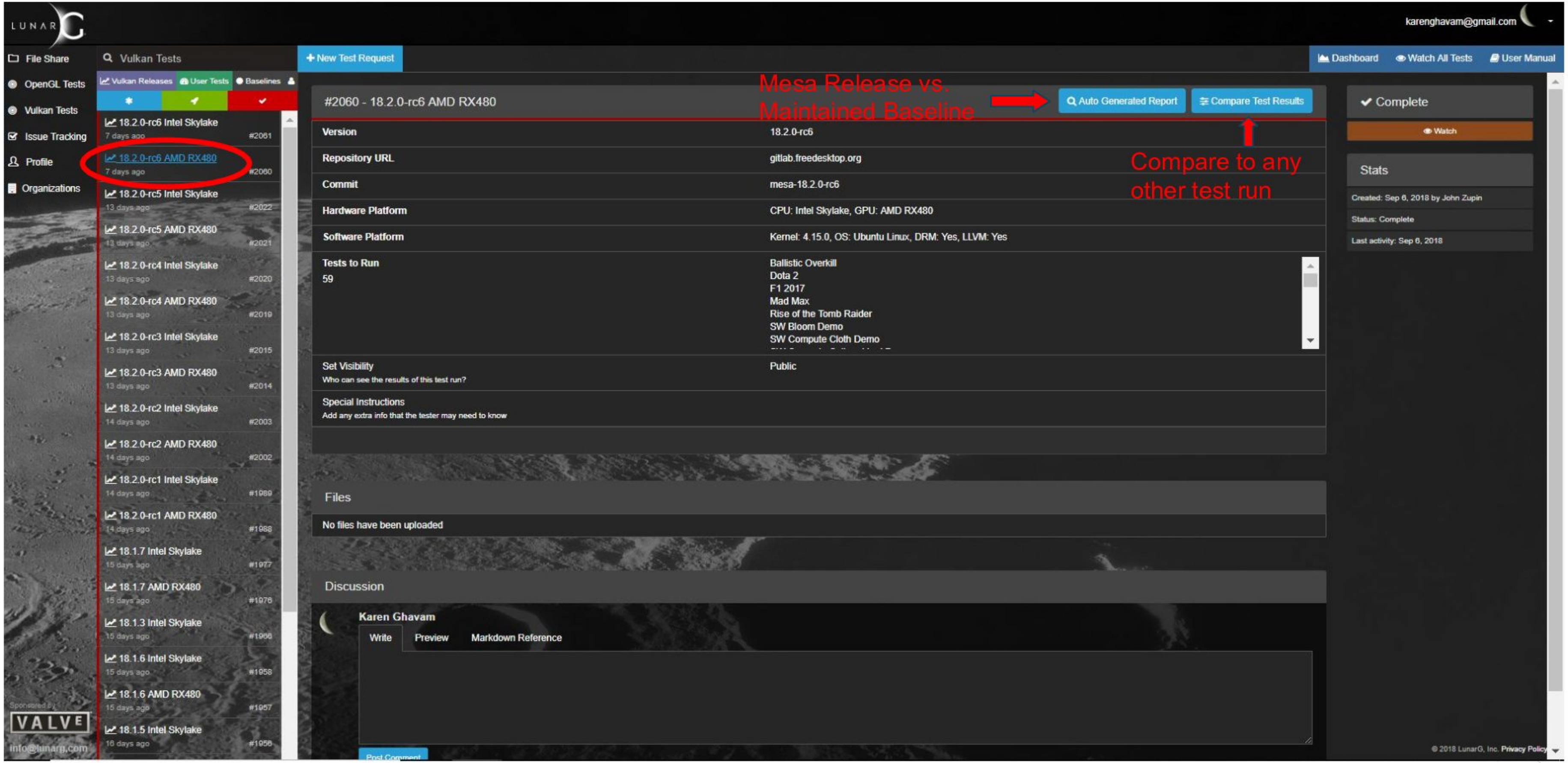
● Public (any test system user)

○ Private (members only)

Special Instructions

Add any extra info that the tester may need to know

View Test Results



Mesa Release vs. Maintained Baseline

Compare to any other test run

Test	Age	Number
18.2.0-rc6 Intel Skylake	7 days ago	#2061
18.2.0-rc6 AMD RX480	7 days ago	#2060
18.2.0-rc5 Intel Skylake	13 days ago	#2022
18.2.0-rc5 AMD RX480	13 days ago	#2021
18.2.0-rc4 Intel Skylake	13 days ago	#2020
18.2.0-rc4 AMD RX480	13 days ago	#2019
18.2.0-rc3 Intel Skylake	13 days ago	#2015
18.2.0-rc3 AMD RX480	13 days ago	#2014
18.2.0-rc2 Intel Skylake	14 days ago	#2003
18.2.0-rc2 AMD RX480	14 days ago	#2002
18.2.0-rc1 Intel Skylake	14 days ago	#1989
18.2.0-rc1 AMD RX480	14 days ago	#1988
18.1.7 Intel Skylake	15 days ago	#1977
18.1.7 AMD RX480	15 days ago	#1976
18.1.3 Intel Skylake	15 days ago	#1966
18.1.6 Intel Skylake	15 days ago	#1958
18.1.6 AMD RX480	15 days ago	#1957
18.1.5 Intel Skylake	16 days ago	#1956

#2060 - 18.2.0-rc6 AMD RX480

Version	18.2.0-rc6
Repository URL	gitlab.freedesktop.org
Commit	mesa-18.2.0-rc6
Hardware Platform	CPU: Intel Skylake, GPU: AMD RX480
Software Platform	Kernel: 4.15.0, OS: Ubuntu Linux, DRM: Yes, LLVM: Yes
Tests to Run 59	Ballistic Overkill Dota 2 F1 2017 Mad Max Rise of the Tomb Raider SW Bloom Demo SW Compute Cloth Demo
Set Visibility Who can see the results of this test run?	Public
Special Instructions Add any extra info that the tester may need to know	

Files

No files have been uploaded

Discussion

Karen Ghavam

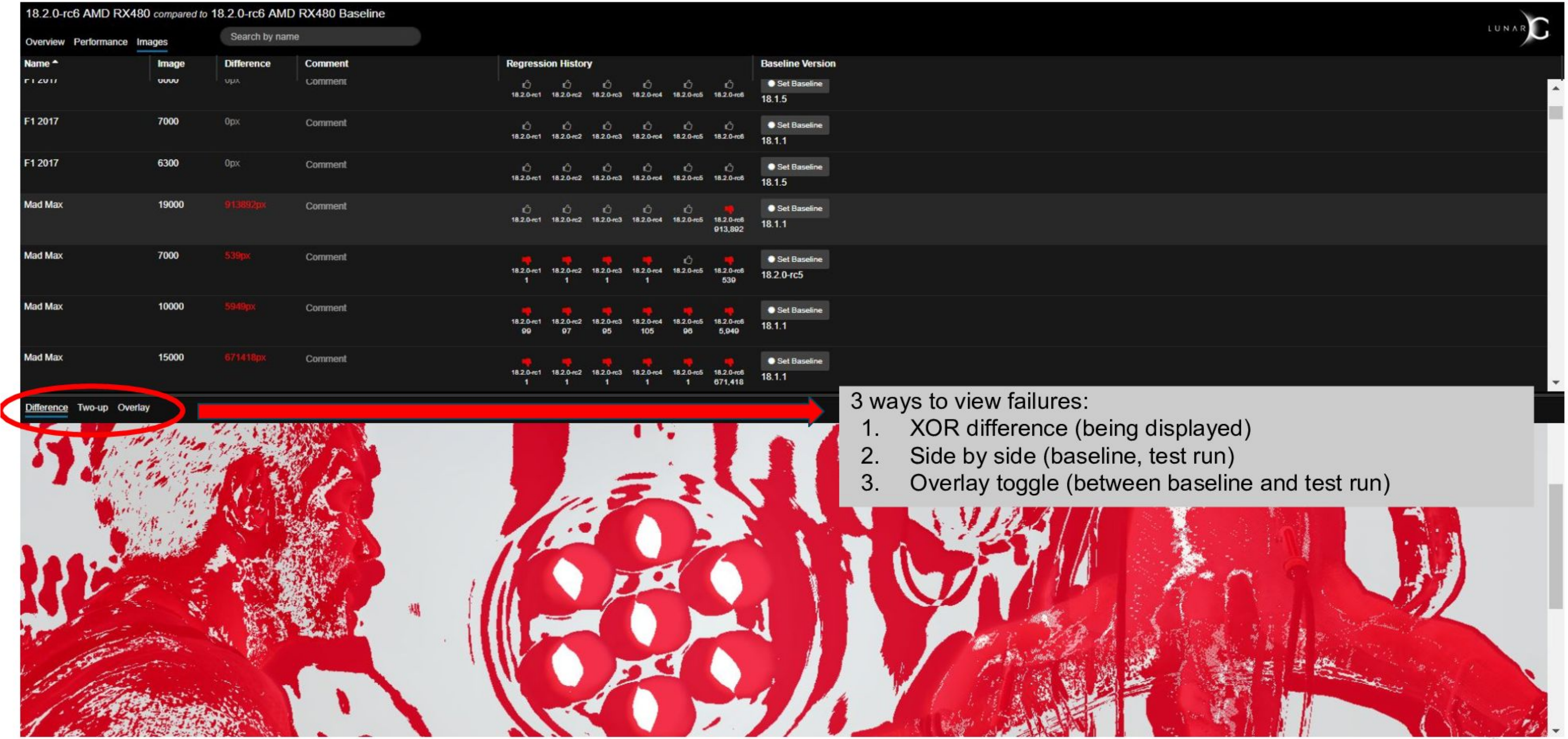
Stats

Created: Sep 6, 2018 by John Zupin

Status: Complete

Last activity: Sep 6, 2018

Image Results



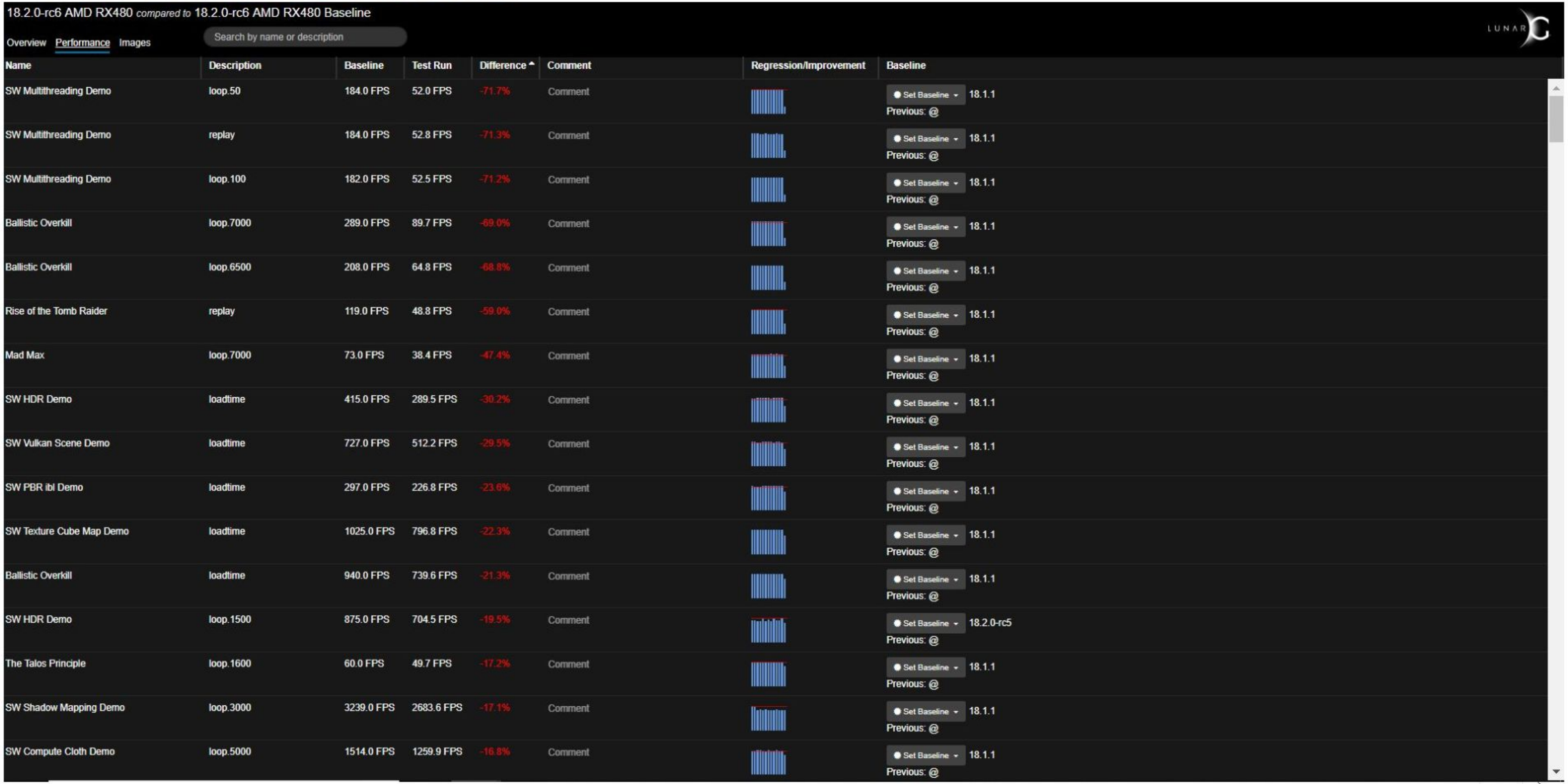
18.2.0-rc6 AMD RX480 *compared to* 18.2.0-rc6 AMD RX480 Baseline

Name ▲	Image	Difference	Comment	Regression History	Baseline Version
F1 2017	6000	0px	Comment	👍 18.2.0-rc1 👍 18.2.0-rc2 👍 18.2.0-rc3 👍 18.2.0-rc4 👍 18.2.0-rc5 👍 18.2.0-rc6	● Set Baseline 18.1.5
F1 2017	7000	0px	Comment	👍 18.2.0-rc1 👍 18.2.0-rc2 👍 18.2.0-rc3 👍 18.2.0-rc4 👍 18.2.0-rc5 👍 18.2.0-rc6	● Set Baseline 18.1.1
F1 2017	6300	0px	Comment	👍 18.2.0-rc1 👍 18.2.0-rc2 👍 18.2.0-rc3 👍 18.2.0-rc4 👍 18.2.0-rc5 👍 18.2.0-rc6	● Set Baseline 18.1.5
Mad Max	19000	913892px	Comment	👍 18.2.0-rc1 👍 18.2.0-rc2 👍 18.2.0-rc3 👍 18.2.0-rc4 👍 18.2.0-rc5 👎 18.2.0-rc6 913,892	● Set Baseline 18.1.1
Mad Max	7000	539px	Comment	👎 18.2.0-rc1 1 👎 18.2.0-rc2 1 👎 18.2.0-rc3 1 👎 18.2.0-rc4 1 👍 18.2.0-rc5 👎 18.2.0-rc6 539	● Set Baseline 18.2.0-rc5
Mad Max	10000	5949px	Comment	👎 18.2.0-rc1 99 👎 18.2.0-rc2 97 👎 18.2.0-rc3 95 👎 18.2.0-rc4 105 👎 18.2.0-rc5 96 👎 18.2.0-rc6 5,949	● Set Baseline 18.1.1
Mad Max	15000	671418px	Comment	👎 18.2.0-rc1 1 👎 18.2.0-rc2 1 👎 18.2.0-rc3 1 👎 18.2.0-rc4 1 👎 18.2.0-rc5 1 👎 18.2.0-rc6 671,418	● Set Baseline 18.1.1

3 ways to view failures:

1. XOR difference (being displayed)
2. Side by side (baseline, test run)
3. Overlay toggle (between baseline and test run)

Performance Results



18.2.0-rc6 AMD RX480 *compared to* 18.2.0-rc6 AMD RX480 Baseline

Name	Description	Baseline	Test Run	Difference ▴	Comment	Regression/Improvement	Baseline
SW Multithreading Demo	loop.50	184.0 FPS	52.0 FPS	-71.7%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Multithreading Demo	replay	184.0 FPS	52.8 FPS	-71.3%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Multithreading Demo	loop.100	182.0 FPS	52.5 FPS	-71.2%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
Ballistic Overkill	loop.7000	289.0 FPS	89.7 FPS	-69.0%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
Ballistic Overkill	loop.6500	208.0 FPS	64.8 FPS	-68.8%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
Rise of the Tomb Raider	replay	119.0 FPS	48.8 FPS	-59.0%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
Mad Max	loop.7000	73.0 FPS	38.4 FPS	-47.4%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW HDR Demo	loadtime	415.0 FPS	289.5 FPS	-30.2%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Vulkan Scene Demo	loadtime	727.0 FPS	512.2 FPS	-29.5%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW PBR ibl Demo	loadtime	297.0 FPS	226.8 FPS	-23.6%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Texture Cube Map Demo	loadtime	1025.0 FPS	796.8 FPS	-22.3%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
Ballistic Overkill	loadtime	940.0 FPS	739.6 FPS	-21.3%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW HDR Demo	loop.1500	875.0 FPS	704.5 FPS	-19.5%	Comment		● Set Baseline ▾ 18.2.0-rc5 Previous: @
The Talos Principle	loop.1600	60.0 FPS	49.7 FPS	-17.2%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Shadow Mapping Demo	loop.3000	3239.0 FPS	2683.6 FPS	-17.1%	Comment		● Set Baseline ▾ 18.1.1 Previous: @
SW Compute Cloth Demo	loop.5000	1514.0 FPS	1259.9 FPS	-16.8%	Comment		● Set Baseline ▾ 18.1.1 Previous: @

Mesa3D Releasing and Testing

- Thanks for your attention
- Questions?