

How does the Mesa3D CI manage to balance fast-paced development and quality of the code?

**By the Gfx CI team
@ Collabora**

Hello!

We're the Mesa3D CI team!

Our team:

Helen, Daniel, Vignesh, Guilherme, Sergi, and David

fdo: @helen.fornazier, @daniels, @vigneshraman, @gallo, @sergi, @dh



COLLABORA

Open First

Today topics:

- What we work with
- Challenges when running the CI
- Tooling and procedures
- Monitoring the CI
- Open questions / Discussion

What we work with

- GitLab CI: pipelines and jobs
- Multiple solutions:
 - Barebone (custom set of scripts connecting various machines)
 - LAVA (Collabora and Lima farm)
 - Tron CI (originally Valve specific runners using containerized env.)
- Different architectures, devices and test suites.
- Kernel from our repository
 - mainline kernel + few custom patches + adjusted defconfig.



Challenges when running the CI

- Scalability & complexity
 - Sharding, parallel, fraction, and the magic of nightly runs
- Infrastructure limitations
 - Finite resources
 - Huge network loads
- **Flakiness and timeouts**
 - At different levels: GitLab, Infra, DUT, and test level
- Reliability of pipeline with about ~250 jobs...
and *Marge* requires all jobs to pass.

Tooling and procedures

- Marge-bot workflow (*pre-merge* tests),
nightly runs and post-merge
- Jobs dependency tree
 - Rules complexity
- Local testing: docker vs rootfs.
- `ci_run_n_monitor + --stress; marge_queue`
- `ci-uprev`
- `Ci-collate`
 - Job, pipeline, patch



Monitoring the CI

- Mesa-performance-tracking
 - ETL model
- Visualization: Grafana
 - Mesa CI Drivers' Performance
 - ci-kdl
 - Mesa CI Quality



Open questions / Discussion

- Increased usage of CI, can we handle it?
- We need more devices. Join us with your home farm!
- Testing different kernels and challenges (missing patches, not mainlined devices yet, differences in fails/flakes)
 - we bundle modules into rootfs & containers... should we split them?
- Booting from NFS and what about the default defconfig? Will our pipeline work with it?
- Testing OpenCL and LLM models?
- Better VA-API testing?
- Better integrate LAVA and Tron CI? Share as much as possible where it's possible (unification/modules/firmware...)



We are hiring
col.la/careers



COLLABORA

Open First