

KernelCI Workshop 2025

August 28th - Amsterdam

Time: 9am - 5pm (Amsterdam time)

Location: Apollo Hotel Amsterdam

Address: Apollolaan 2

1077 BA Amsterdam, NL

Google maps: <https://maps.app.goo.gl/j9KFQV7FDAb3TSG37>

Attendees

Local:

Kevin Hilman (BayLibre)

Sjoerd (Collabora)

Ben Copeland (linaro)

Pawel (Collabora)

Mark Brown

Drew Fustini ([Beagleboard.org](https://beagleboard.org), Tenstorrent)

Lothar Rubusch ()

Chris Fiege (Pengutronix)

Fabian Pflug (Pengutronix)

Chris Paterson (Renesas/CIP)

Gustavo Padovan (Collabora)

Remote:

Jeny Sadadia (Collabora)

Denys Fedoryshchenko (Collabora)

Tales (RH)

Timetable:

11:00 Coffee!

13:00 Lunch

15:00 Coffee

Overview

Organization updates

- Paying members growth
 - TI joined

- Linaro joined
 - 2 more “arm”/embedded companies hopefully joining
- new Project Manager from the LF - Yasmeen
 - Seen as helpful for the project
 - Taking care of financials and other logistics
- Structuring technical oversight and working groups
 - Infrastructure WG created
 - TSC elections coming up
 - Call for self-nomination coming up soon (next couple of weeks)
 - 1 year term; 4 slots initially + infrastructure lead (currently Denys)
- ProFUSION helping Dashboard and DB development as supplier
- Onboarding new contributors
 - Seeing interest from multiple people/companies to start actually contributing
- Future: look at marketing/outreach activities
 - Look at more workshops to drive involvement
 - Outreachy/summer of code

Architecture overview

- Go over <https://docs.kernelci.org/intro/architecture/>
- Current challenges
 - KCIDB transition
 - quality of the Maestro -> Dashboard pipeline
 - staging environment

Test results evaluation and reporting

- Biggest frontier is to clean up/filter out the good data and report those regressions to the community

References

- [Mailing list discussion](#)
- Lab health [dashboard](#)

Discussions topics

- Maintainer’s usecases: discuss and prioritize key usecases and understand the steps to implement them.
 - Kevin: Not really using it at the moment; Hard to define a focus on the results he cares about.
 - speed of testing is important. Lack of information of turnaround time

- Few hours / half a day is ok; For mark it's often overnight
 - KernelCI testing "too much". Need clear input from maintainers on what to test
 - For things where rapid turn-around is required both Kevin & Mark run locally
 - Mark currently uses mostly the dashboard at the moment; tooling setup can be a barrier
 - AI: prioritize next, mainline and stable(-rc)
 - Linaro is testing stable kernels and reporting; Main thing to move would be the report generation
 - Potentially starting with the data that gets pushed by Linaro, expanding it over time to collect the other data as well.
 - AI: Linaro to look at it in the dashboard
 - Pull request testing
 - Replicating Intel 0-day features
 - Bisection support
 - as a standalone VM/k8s that can run separately
 - Separate out running bisection from decisions on what to bisect with an API (also retrieve the bisect logs)
 - Useful to annotate logs with links to KCIDB results (example from broonie's local automation):
 - <https://lore.kernel.org/all/95449490-fa58-41d4-9493-c9213c1f2e7d@sirena.org.uk/>
 -
- Stable-rc & linux-next: How to create and deliver unified reports to the stable maintainers in a timely manner
 - Greg would like to a single report from e.g. kernelci combining inputs from all different test systems
 - Both positive and negative feedback in a timely manner
 - Should compare with previous runs (e.g. to detect a category of tests are missing, or some infrastructure is missing)
 - Needs to potentially be catered at the relevant maintainers (e.g. just subsystems, more higher level overview)
 -
- quality of test results - we run many thousands of tests everyday, but the confidence level is not where we want it to be. Some tests and platforms are unstable, but currently we have no tracking of that.
 - logspec: https://github.com/kernelci/logspec/blob/main/tests/test_baseline.py
 - Need to make it clear when there is an infrastructure failure
 - There "must" be a tool already out there being used to help parse boot/test results etc. Perhaps worth an investigation rather than re-inventing the wheel
 - Loki...
 - sample of test results
<https://gist.github.com/padovan/801d8a5662d262fe9155d6aa18df7a6c>
 - Labs working group might be relevant

- Mainline mandates the standard
- AI: Kernelci should make sure it tests tagged release (e.g. RC2 test)
 - <https://github.com/kernelci/kernelci-pipeline/issues/1292>
- AI: document prototyping env with django mgmt scripts
- Labs:
 - AI: start the Labs WG
 - infra issues [dashboard](#)
 - CIP lab qemu issues ([example](#)) are caused by one of the workers not having docker jobs enabled
 - Goal of the WG to discuss the Labs integration
 - look at reliability issues
 - Design labgrid integration
 - improve LAVA & labgrid. Also CI Ecosystem labs (eg TI, Qualcomm)
 - Checklist for hardware manufacturers to make their hardware testable (ptx has an internal list they might share)
 - Lab usage efficiency improvements (more of a Maestro architecture, but affects lab usage):
 - Job prioritisation.
 - Job dependency usage (eg, don't try things that need a NFS root filesystem if NFS boot didn't work).
 - Also depending on a baseline (e.g. RC2 kernel) as a dependency for the maintainer tree
 - Deduplication of tests/builds for the same commit (especially an issue at -rc1).
 - Chris: Status of a Lab: How can we see how our lab is doing?
 - From an Infrastructure perspective
 - From a customer's perspective: How can we see how a single device is doing in a lab?
- What's the sweet spot for email notifications? It is known that bots send too many emails. We don't want to repeat that with KernelCI. Our notifications infra is still PoC, so let's discuss the next steps.
 - Have kernelci respond to pull request
 - Need to ensure filtering out on stuff the maintainer cares about (e.g. not regressions outside of the pull request)
 - Zero-day has similar issues by being too broad in the report
 - Maintainers should be able to define what they care about
 - People getting pull requests in have different questions to the people sending the PR.
- KCIDB-ng and Dashboard DB roadmap: we have been revamping how we store test results sent to KCIDB, so this is a pretty good discussion topic as well.
 - [KCIDB Roadmap](#)
 - <https://github.com/kernelci/kernelci-pipeline/issues/1293>
- Maestro

- AI: improve timeout setting using moving average.
- Defining Hardware/Platform information: Today hw info is pretty much non-existent in KernelCI and a big limitation for showing results filtered by hw capabilities.
 - Architecture
 - Vendor
 - Board
 - Device-tree - can be used to mine a lot of info
 - MVP: OEM/SOM/Soc as well as revision
 - CPU family (just what is a family)
 - Expansion cards?
 - Extra information: u-boot, firmware, TF-A etc (runtime)
 - Lab WG: Add some kind of guidelines for Lab admins (e.g. how to differentiate device types)
 - AI: KCI-db/Maestro to define the schema of required information (ticket to be opened)
- Web Dashboard: what are we missing? What are the next steps?
 - Lab filtering
- Testing patches from the mailing list. What to look for in this testing?
- Listening to test events in the Maestro API: current limitations and wishlist to improve support the CI ecosystem that want to test jobs generated by Maestro
- kci-dev: requesting ad-hoc test execution to test patches against regressions, bisections, etc.
- bisection support: how to do it?
 - Mark has some tooling already; needs to be adapted to e.g. kci-dev
 - Kci-dev has a command to run a bisection as well
 - Ben has a similar script as well
 - Usage of `git bisect` requires a local checkout (used to have a VM dedicated just for that)
- Test results retention policy: for how long should we retain results? Which summary info should we keep?
 - prevent storage bloat etc.
 - Currently saving everything for 6 months
 - Should be defined per kernel tree (e.g. next is less important)
 - Data could be more focussed
 - pass/fail data for longer time might be of interest, but not always the detailed logging
 - Some data e.g. build results is reproducible
 - AI: Denys to have a first look into a policy
 - Possible a service inside kcidb-ng for purge logs and db data
- KernelCI sysadmin policies
 -  KernelCI Sysadmin Policy
- User accounts/tokens
 - The Dashboard can be a holder of the Users UX
 - However, we need to think about how and who does the authentication

- AI: Reach out to Mike Dolan to ask how/who at the LF can help with
- Should the lab configuration be higher level (more like the boards/tests mapping we had with v1)?
- Onboarding documentation/tooling:
 - Programming the API (plus general API docs).
 - Command line tooling (kci-dev is pretty good here, but not well signposted from the docs).
 - Configuration?
 - AI: collect more feedback
- Introspection/debuggability of configuration improvements (Denys is already doing a bunch of work here).
- Qemu testing:
 - Look at integrating Tuxtest as another potential lab
 - Easier to work with for virtual testing then lava
 - Should also be able to test virtualized ARM eventually

Interesting Links:

- USB I2C Interface: <https://www.fischl.de/i2c-mp-usb/>