Merging an existing framework into KernelCI

How to test your own kernel project with KernelCI

Who am I:

Alice Ferrazzi

Gentoo:
- Gentoo Kernel Project Leader
- GKernelCI creator

KernelCI:
- TSC (Technical Steering Committee) member

CIP (Civil Infrastructure Platform):
- CIP Testing Working Group member

MIRACLE LINUX powered by Cybertrust Japan Co., Ltd.
- Software Engineer
- EMlinux embedded Linux distribution - Lead CI system development

summary

- KernelCI
  - What's KernelCI?
  - Who is doing it
  - Why KernelCI is needed
  - KernelCI composition
  - Big picture
  - Current available KernelCI test labs
  - Framework definition
  - Merging a testing framework into KernelCI
- KernelCI native implementation
  - What is KernelCI native implementation
  - KernelCI native implementation
  - CIP (Civil Infrastructure Platform)
  - Merging CIP testing framework into KernelCI native
  - CIP testing framework
  - CIP testing framework merged into KernelCI
  - KernelCI currently supported tests
  - CIP KernelCI results email example
  - Something broke → Regression email
  - CIP Web Dashboard
- Conclusion

What’s KernelCI?

KernelCI is a community-based open source distributed test automation system focused on upstream kernel development

Currently testing upstream kernel on 155 of physical boards and virtual boards

Who is doing it

- TSC (Technical Steering Committee)
  - Formed by KernelCI core developers and maintainer
  - KernelCI development and mantainance
- Advisory Board
  - Premium organizations representatives involved in KernelCI
  - Manage budgets and help coordinating tasks

Premium Members

General Members

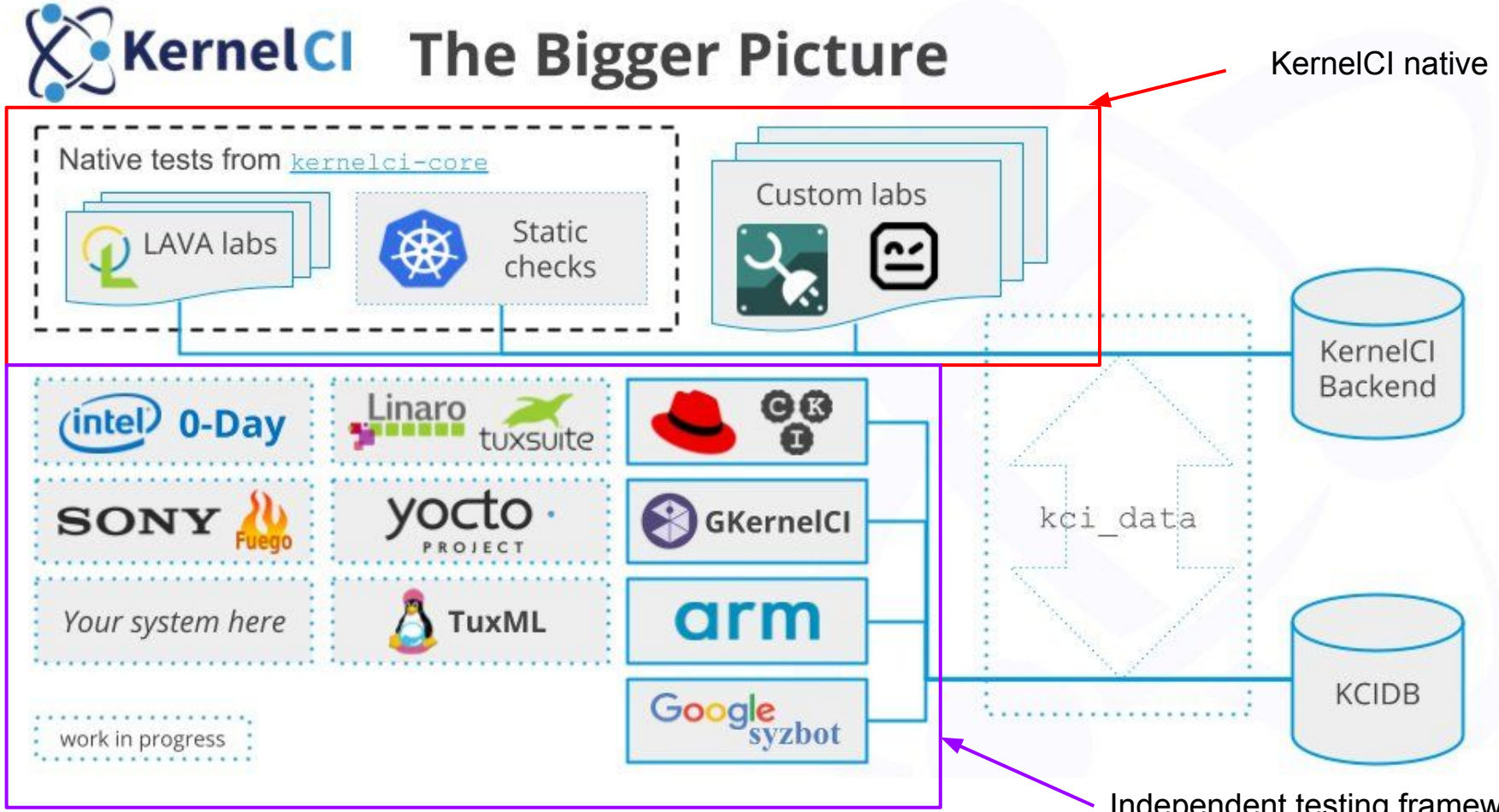
Why KernelCI is needed

To ensure the quality, stability and long-term maintenance of the Linux kernel by maintaining an open ecosystem around test automation practices and principles

KernelCI composition

- KernelCI-core core-tools
  - The main configuration and tools of KernelCI.
- Backend (currently rework in progress)
  - Provides the KernelCI web API https://api.kernelci.org/
- Frontend
  - Web dashboard showing the data available from the backend
- Jenkins
  - Orchestrate builds and tests
- Test labs (currently mostly LAVA)
  - Test-definitions
    - Keep lava jobs test definitions
    - You add code here if you want to add new lava jobs tests to KernelCI
  - Lava-docker
    - For making your own KernelCI LAVA testing laboratory (works with docker)
- KCIDB
  - Tool to submit kernel test data

KernelCI The Bigger Picture



Current available KernelCI test labs

- Lab-baylibre
- Lab-broonie
- Lab-cip
- Lab-clabbe
- Lab-collabora
- Lab-collabora-staging
- Lab-kontron
- Lab-linaro-lkft
- Lab-mhart
- Lab-nxp
- Lab-pengutronix
- Lab-theobroma-systems
- **Add your lab here!**

Framework definition

Testing framework including Kernel building, booting and testing code

For example the CIP (Civil Infrastructure Platform) project have its own testing framework for testing the CIP SLTS kernel tree.

Merging a testing framework into KernelCI

There are two main ways of adding your work into KernelCI:

- KernelCI native (Talked in this presentation)
  - Merging kernel testing code
  - for example, in the next slide we will explain how we merged CIP framework code into KernelCI native
- KCIDB (Presented at LCA 2021 in the GkernelCI presentation)
  - Sharing kernel testing results
  - For example, we will explain how we manage to send Gentoo Linux kernel tests results into KernelCI common database

KernelCI native implementation

What is KernelCI native implementation

- Main testing framework of KernelCI
- Automate build, boot and test (bisection experimental) of kernel trees
- Developed and maintained by the KernelCI community

KernelCI native implementation

Pros

- Already integrated kernel boot and testing framework
- KernelCI API support
- Maintained by the KernelCI community
- Linux Foundation project
- Can use KernelCI test laboratory resources (if the code get merged into KernelCI native)

Cons

- Made only for kernel testing. Task that are out of the scope of KernelCI native are not integrated into the KernelCI native upstream code. (For example distribution packaging)

CIP (Civil Infrastructure Platform)

CIP is a Linux Foundation project that aims to establish a “base layer” of industrial-grade tooling using the Linux kernel and other open source projects.

https://www.cip-project.org/

Merging CIP testing framework into KernelCI native

- CIP team decided to merge the current CIP testing framework into KernelCI native because of the KernelCI native pros
- In the next slides, we will explain what is CIP and how CIP managed to merge it

CIP testing framework

- Maintained by the CIP community
- Few code reuse
- Future changes and new features are done by the CIP community
- Results are only shared to the CIP team on the GitLab pipeline
- Testing on the CIP lava laboratory (21 physical and virtual active boards under test)

CIP testing framework merged into KernelCI

- Maintained both by CIP community and KernelCI
- Code reuse
- Integrate future changes and futures from the KernelCI community
- Results are integrated on KernelCI and results summary are shared to the CIP mailing list
- Testing on all the KernelCI laboratory (around 5000 test results and 190 builds on 155 physical and virtual boards)
- Experimental bisection

KernelCI currently supported tests

- Kselftest (cpufreq, filesystem, futex, lib, livepatch, lkdtm, rtc, seccomp)
- LTP (crypto, fcntl-locktests, ima, ipc, mm, pty, timers)
- Preempt-rt
- Sleep
- Cameras with lc-compliance and v4l2-compliance (uvc, vivid)
- SMC (Spectre Meltdown Checker)
- USB
- IGT GPU (amd, i915, panfrost)
- IGT KMS (exynos, rockchip, tegra)
- **Add your tests**

CIP KernelCI example summary results emails

cip/linux-5.10.y-cip build: 184 builds: 3 failed, 181 passed, 4 errors, 10 warnings (v5.10.83-cip1) #20112

KernelCI bot

Dec 5 #20112

cip/linux-5.10.y-cip build: 184 builds: 3 failed, 181 passed, 4 errors, 10 warnings (v5.10.83-cip1)

Full Build Summary: https://kernelci.org/build/cip/branch/linux-5.10.y-cip/kernel/v5.10.83-cip1/

Tree: cip
Branch: linux-5.10.y-cip
Git Describe: v5.10.83-cip1
Git Commit: 2332f07a324fd78d7c7436deeed23cd7db441ea7
Git URL: https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git
Built: 7 unique architectures

If something broke → regression email

cip/linux-4.19.y-cip baseline: 121 runs, 1 regressions (v4.19.217-cip62) #kernelci

Date ▲ 1 - 3 of 3 < >

kernelci.org bot <bot@...> Nov 30 #7025

cip/linux-4.19.y-cip baseline: 121 runs, 1 regressions (v4.19.217-cip62)

Regressions Summary
---

platform	arch	lab	compiler	defconfig	regressions
panda	arm	lab-collabora	gcc-10	omap2plus_defconfig	1

Details: https://kernelci.org/test/job/cip/branch/linux-4.19.y-cip/kernel/v4.19.217-cip62/plan/baseline/

Test: baseline
Tree: cip
Branch: linux-4.19.y-cip
Describe: v4.19.217-cip62
URL: https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git
SHA: dc62e26e3be875a7324b85b8274c13a335e610dd

Test Regressions
---

platform	arch	lab	compiler	defconfig	regressions
panda	arm	lab-collabora	gcc-10	omap2plus_defconfig	1

Details: https://kernelci.org/test/plan/id/61a587a0ab3b0079bd18f6d7

Results: 5 PASS, 1 FAIL, 0 SKIP
Full config: omap2plus_defconfig
Compiler: gcc-10 (arm-linux-gnueabihf-gcc (Debian 10.2.1-6) 10.2.1 20210110)
Plain log: https://storage.kernelci.org//cip/linux-4.19.y-cip/v4.19.217-cip62/arm/omap2plus_defconfig/gcc-10/lab-collabora/baseline-panda.txt
HTML log: https://storage.kernelci.org//cip/linux-4.19.y-cip/v4.19.217-cip62/arm/omap2plus_defconfig/gcc-10/lab-collabora/baseline-panda.html
Rootfs: http://storage.kernelci.org/images/rootfs/buildroot/kci-2020.05-6-g8983f3b738df/armel/baseline/rootfs.cpio.gz

* baseline.dmesg.emerg: https://kernelci.org/test/case/id/61a587a0ab3b0079bd18f6dd
  new failure (last pass: v4.19.216-cip61)
  2 lines

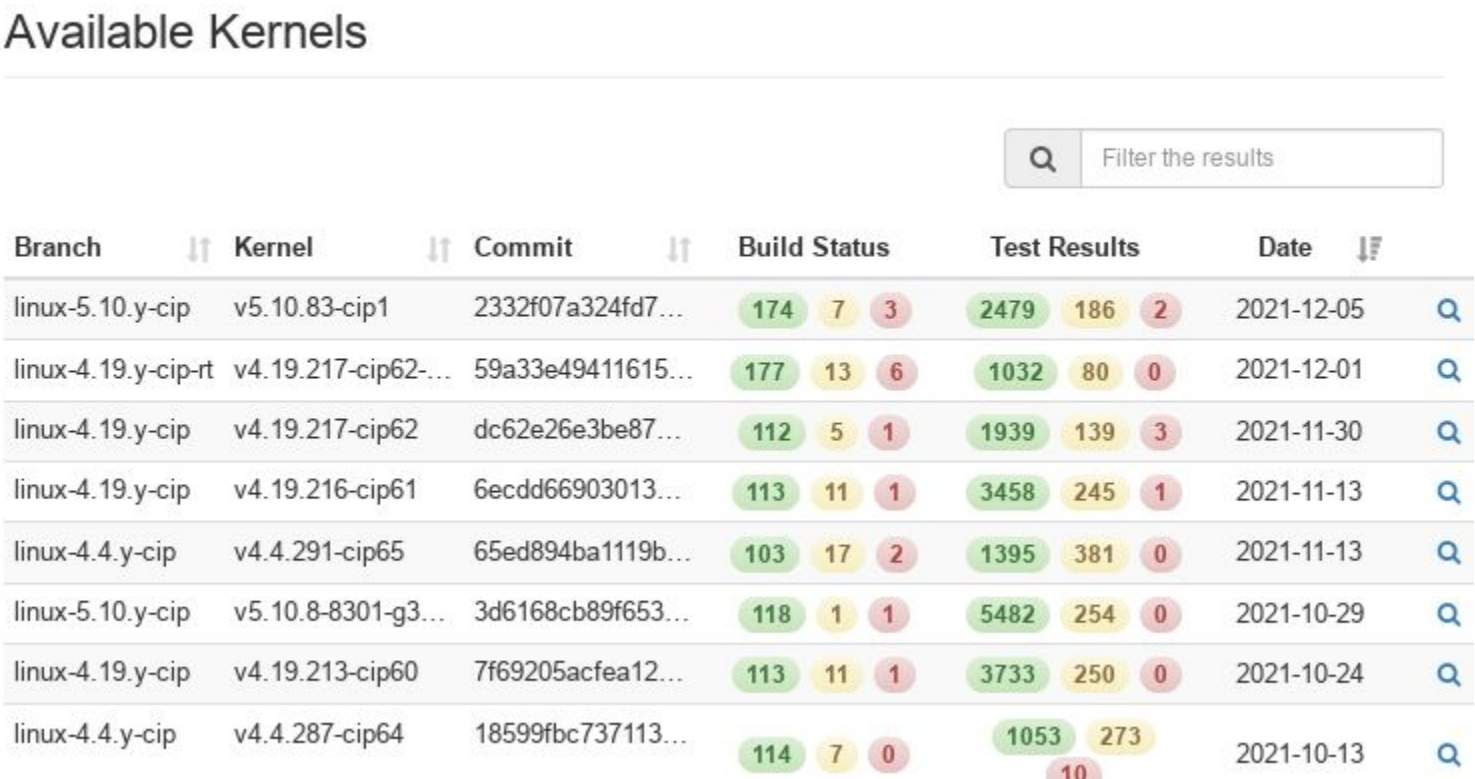
2021-11-30T02:08:20.120053 kern :emerg : BUG: spinlock bad magic on CPU#0, udevd/110
2021-11-30T02:08:20.129545 kern :emerg : lock: emif_lock+0x0/0xffffecfc [emif], .magic: dead4ead, .owner: <none>/-1, .owner_cpu: -1
2021-11-30T02:08:20.143995 <8>[21.198120] <LAVA_SIGNAL_TESTCASE TEST_CASE_ID=emerg RESULT=fail UNITS=lines MEASUREMENT=2>

Log files

Last success run

Relevant error message

CIP Web Dashboard https://cip.kernelci.org



Conclusion

Working on KernelCI native helps to create a shared set of tooling and infrastructures for improving Kernel dependability and assurance.
KernelCI native can be a useful tool for discovering kernel regressions bugs.

KernelCI documentation

KernelCI native CIP test integration project

KernelCI Maintainers Channels

- IRC
  - #KernelCI on libera.chat
- Slack
  - https://kernelci.slack.com/
- Twitter
  - https://twitter.com/kernelci
- Groups.io
  - https://groups.io/g/kernelci