KernelCI

Kernel CI – How Far Can It Go?

Embedded Open Source Summit 2023

Nikolai Kondrashov

KernelCI

Nikolai Kondrashov

spbnick

Sr. Software Engineer, Red Hat, CKI Project
KCIDB developer & maintainer, KernelCI
Electronics and embedded enthusiast

Born in Russia, living in Finland

Contents

- Kernel Testing Systems
- What are KernelCI and KCIDB?
- What's Continuous Integration?
- The Metrics of CI
- Where are we with upstream Kernel's CI?
- Hard Limits
- Challenges
- What Can We Do?

KernelCI

Kernel testing systems

Kernel testing systems

...

Kernel testing systems

```
Results from Linaro's test farm.
No regressions on arm64, arm, x86_64, and i386.

Summary
------------------------------------------------------------------------


kernel: 5.7.13-rc3
git repo: https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable-rc.git
git branch: linux-5.7.y
git commit: d3223abaf6fdd20e0894b357a0f7f1da21a29226
git describe: v5.7.12-117-gd3223abaf6fd
Test details: https://qa-reports.linaro.org/lkft/linux-stable-rc-5.7-oe/build/v5.7.12-117-gd3223abaf6fd

No regressions (compared to build v5.7.12)

No fixes (compared to build v5.7.12)


Ran 34339 total tests in the following environments and test suites.

Environments
--------------
- dragonboard-410c
- hi6220-hikey
- i386
- juno-r2
- juno-r2-compat
- juno-r2-kasan
- nxp-ls2088
- qemu_arm
- qemu_arm64
- qemu_i386
- qemu_x86_64
- x15
- x86
- x86-kasan

Test Suites
```

KMSAN: kernel-usb-infoleak in hif_usb_send

[Date Prev][Date Next][Thread Prev][Thread Next][Date Index][Thread Index]

- *To*: glider@xxxxxxxxxx, gregkh@xxxxxxxxxxxxxxxxxxx, ingrassia@xxxxxxxxxxxxxxx, linux-kernel@xxxx
- *Subject*: KMSAN: kernel-usb-infoleak in hif_usb_send
- *From*: syzbot <syzbot+f83a1df1ed4f67e8d8ad@xxxxxxxxxxxxxxxxxxxxxxxxxxx>
- *Date*: Tue, 11 Aug 2020 10:06:18 -0700

```
Hello,

syzbot found the following issue on:

HEAD commit:    ce8056d1 wip: changed copy_from_user where instrumented
git tree:       https://github.com/google/kmsan.git master
console output: https://syzkaller.appspot.com/x/log.txt?x=124d453c900000
kernel config:  https://syzkaller.appspot.com/x/.config?x=3afe005fb99591f
dashboard link: https://syzkaller.appspot.com/bug?extid=f83a1df1ed4f67e8d8ad
compiler:       clang version 10.0.0 (https://github.com/llvm/llvm-project/ c2443155a0fb245c8f17f2c1c72b6ea391e86e81)
userspace arch: i386

Unfortunately, I don't have any reproducer for this issue yet.

IMPORTANT: if you fix the issue, please add the following tag to the commit:
Reported-by: syzbot+f83a1df1ed4f67e8d8ad@xxxxxxxxxxxxxxxxxxxxxxxxxxx

usb 4-1: ath9k_htc: Transferred FW: ath9k_htc/htc_9271-1.4.0.fw, size: 51008
==========================================================
BUG: KMSAN: kernel-usb-infoleak in kmsan_handle_urb+0x20/0x40 mm/kmsan/kmsan_hooks.c:307
CPU: 1 PID: 13012 Comm: kworker/1:6 Not tainted 5.8.0-rc5-syzkaller #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011
Workqueue: events request_firmware_work_func
Call Trace:
```

stable-rc/linux-4.9.y build: 197 builds: 1 failed, 196 passed, 4 warnings (v4.9.232)

kernelci.org bot bot at kernelci.org
Sat Aug 1 08:24:58 UTC 2020

- Previous message: [PATCH stable] MIPS: Loongson: Introduce and use loongson_llsc_mb()
- Next message: stable-rc/linux-4.9.y build: 197 builds: 1 failed, 196 passed, 4 warnings (v4.9.232)
- **Messages sorted by:** [date] [thread] [subject] [author]

```
stable-rc/linux-4.9.y build: 197 builds: 1 failed, 196 passed, 4 warnings (v4.9.232)

Full Build Summary: https://kernelci.org/build/stable-rc/branch/linux-4.9.y/kernel/v4.9.232/

Tree: stable-rc
Branch: linux-4.9.y
Git Describe: v4.9.232
Git Commit: 8d6b541290cb92093bd2a7bb00c1d58d01abe183b
Git URL: https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable-rc.git
Built: 6 unique architectures

Build Failure Detected:

mips:
    32r2el_defconfig: (gcc-8) FAIL

Warnings Detected:

arc:
```

✅ PASS: Test report for kernel 5.7.12-3ff3d4f.cki (stable-queue)

CKI Project cki-project at redhat.com
Tue Aug 4 12:35:15 UTC 2020

- Previous message: fixing 4.14-stable's broken DM cache writethrough support [was: Re: [(resend) PATCH v3: {linux-4.14.y}] dm cache: submit writethrough writes in parallel to origin and cache]
- Next message: stable-rc/linux-4.4.y build: 190 builds: 1 failed, 189 passed, 2 errors, 13 warnings (v4.4.232-30-g52247eb98ebe)
- **Messages sorted by:** [date] [thread] [subject] [author]

```
Hello,

We ran automated tests on a recent commit from this kernel tree:

       Kernel repo: https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable-rc.git
            Commit: 3ff3d4f43856 - x86/i8259: Use printk_deferred() to prevent deadlock

The results of these automated tests are provided below.

    Overall result: PASSED
             Merge: OK
           Compile: OK
             Tests: OK

All kernel binaries, config files, and logs are available for download here:

  https://cki-artifacts.s3.us-east-2.amazonaws.com/index.html?prefix=datawarehouse/2020/08/03/611391

Please reply to this email if you have any questions about the tests that we
ran or if you have any suggestions on how to make future tests more effective.

        ,-.   ,-.
       ( C ) ( K )  Continuous
        `-',-.-'    Kernel
          ( I )     Integration
           `-'
```

Re: [PATCH V4] hwmon: add fan/pwm driver for cors

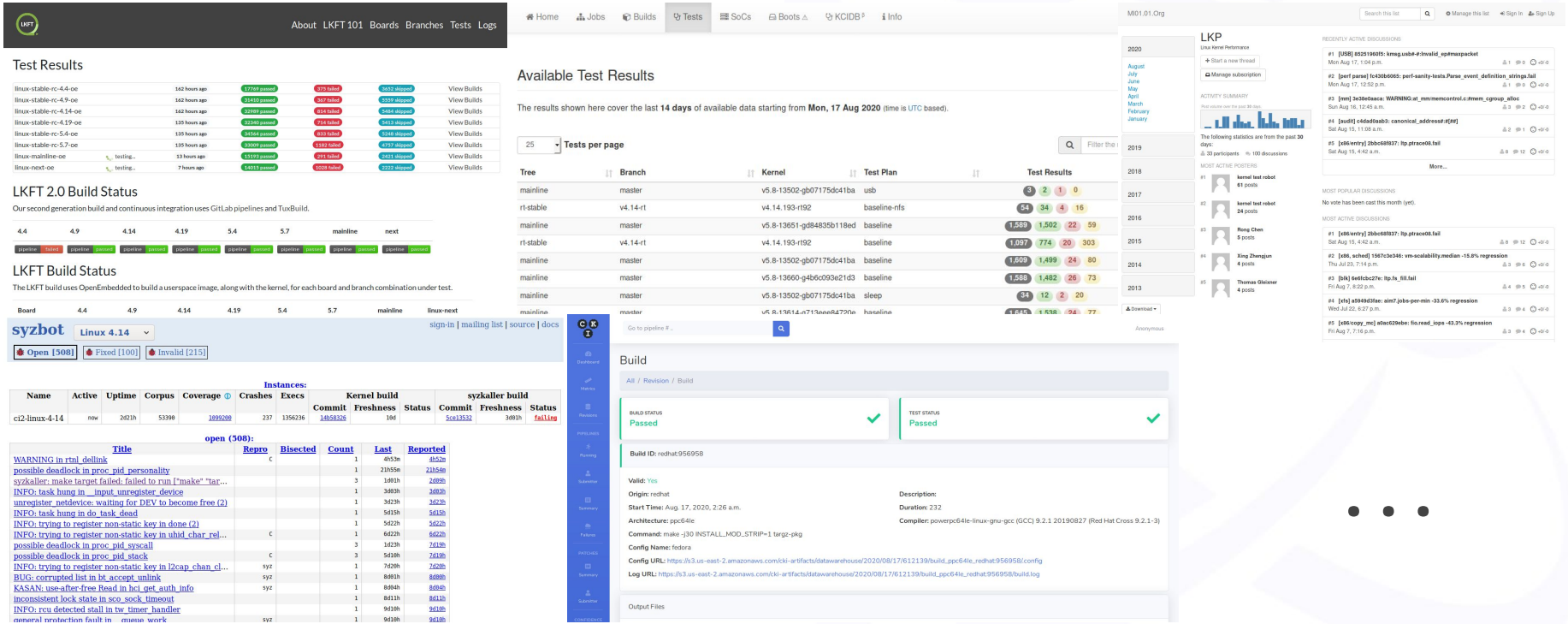
[Date Prev][Date Next][Thread Prev][Thread Next][Date Index][Thread Index]

- *To*: jaap aarts <jaap.aarts1@xxxxxxxxx>, Jean Delvare <jdelvare@xxxxxxxx>, Guenter Roeck <linux@xxxxxxx
usb@xxxxxxxxxxxxxx>, Barnabás Pőcze <pobrn@xxxxxxxxxxxxx>
- *Subject*: Re: [PATCH V4] hwmon: add fan/pwm driver for corsair h100i platinum
- *From*: kernel test robot <lkp@xxxxxxxxx>
- *Date*: Sun, 16 Aug 2020 17:34:40 +0800
- *Cc*: kbuild-all@xxxxxxxxxxxx, jaap aarts <jaap.aarts1@xxxxxxxxx>
- *In-reply-to*: <20200815211617.86565-1-jaap.aarts1@gmail.com>
- *References*: <20200815211617.86565-1-jaap.aarts1@gmail.com>
- *User-agent*: Mutt/1.10.1 (2018-07-13)

```
Hi jaap,

Thank you for the patch! Perhaps something to improve:

[auto build test WARNING on hwmon/hwmon-next]
[also build test WARNING on v5.8 next-20200814]
[If your patch is applied to the wrong git tree, kindly drop us a note.
And when submitting patch, we suggest to use '--base' as documented in
https://git-scm.com/docs/git-format-patch]

url:    https://github.com/0day-ci/linux/commits/jaap-aarts/hwmon-add-fan-pwm-driver-for-corsair-h100i-platinum/20200816-085929
base:   https://git.kernel.org/pub/scm/linux/kernel/git/groeck/linux-staging.git hwmon-next
config: nios2-randconfig-c003-20200816 (attached as .config)
compiler: nios2-linux-gcc (GCC) 9.3.0
```

KernelCI

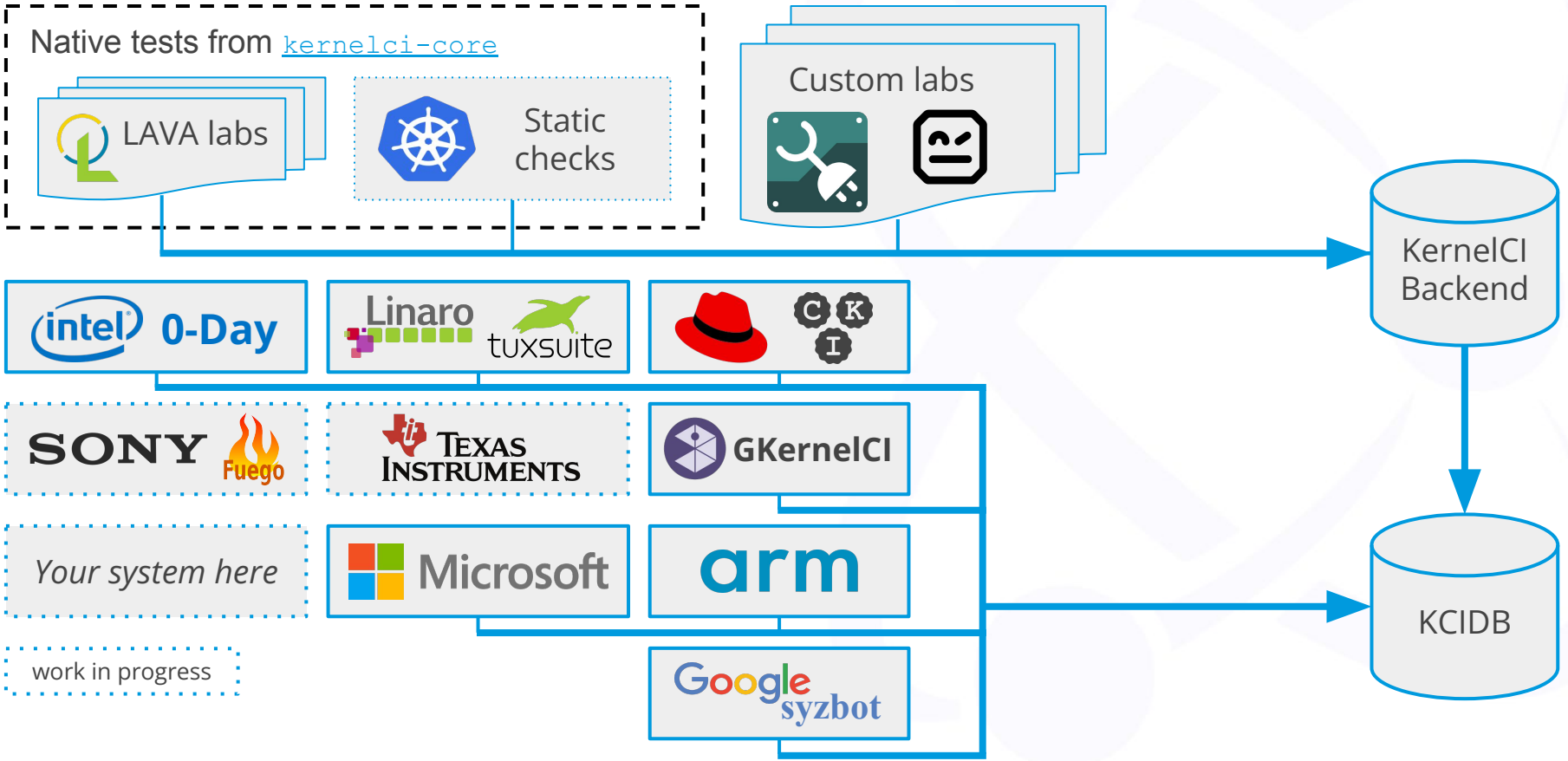
Kernel testing systems



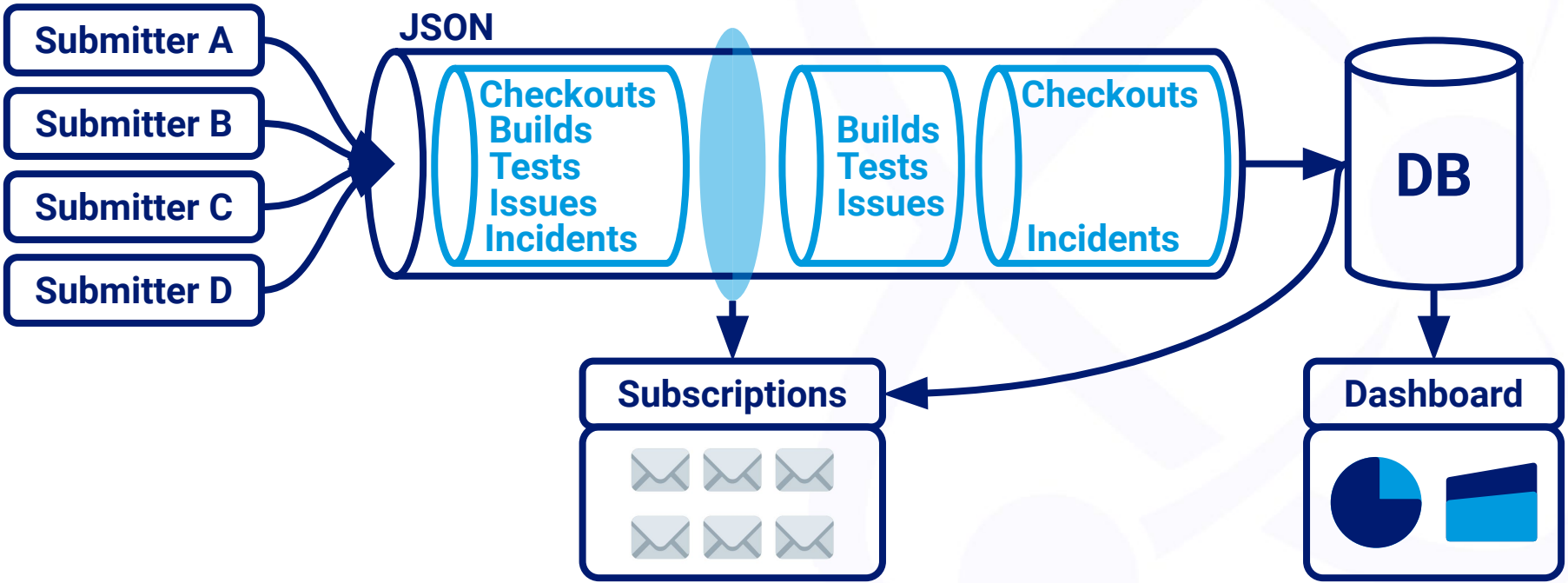
KernelCI

What are KernelCI and KCIDB

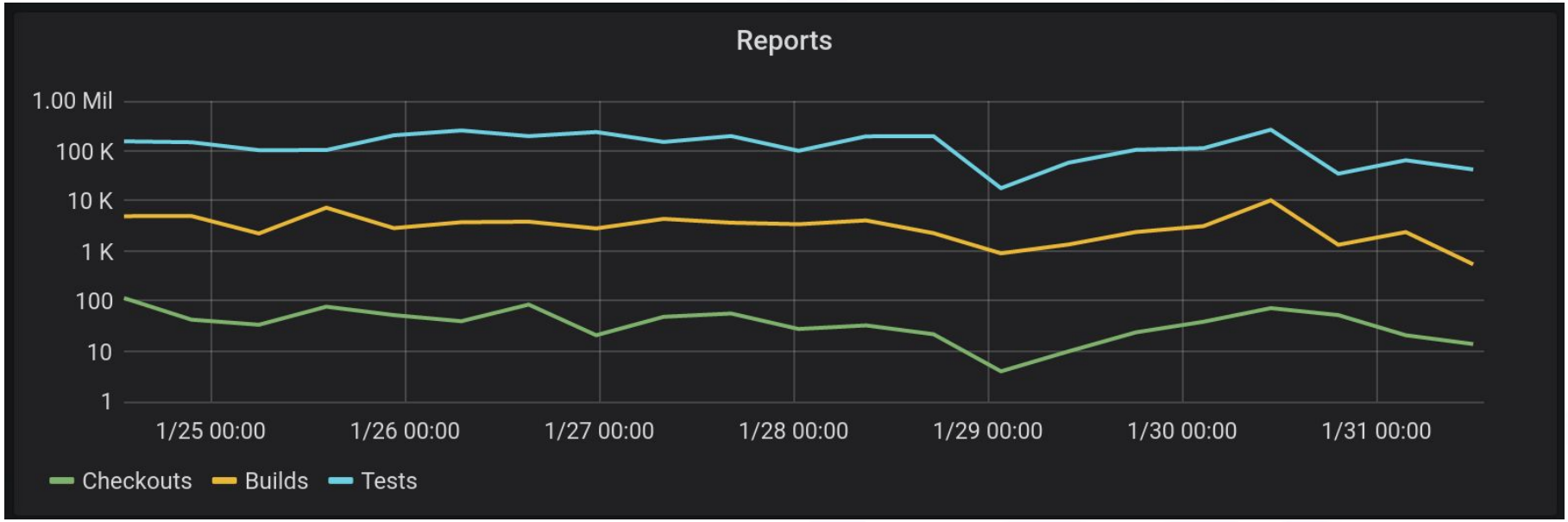
What are KernelCI and KCIDB



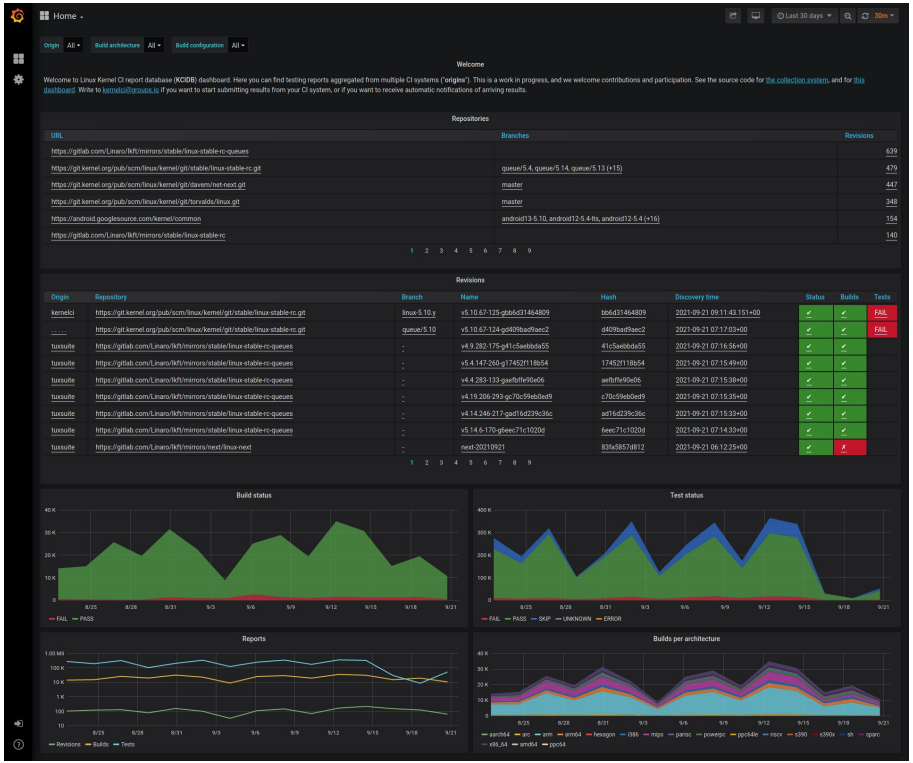
What are KernelCI and KCIDB

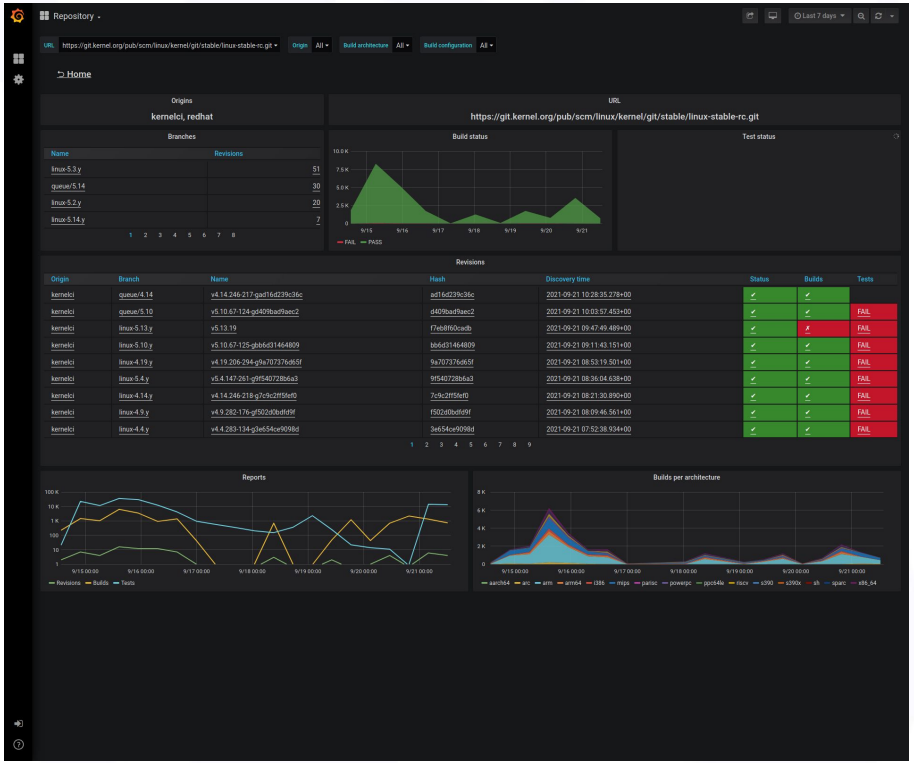


What are KernelCI and KCIDB

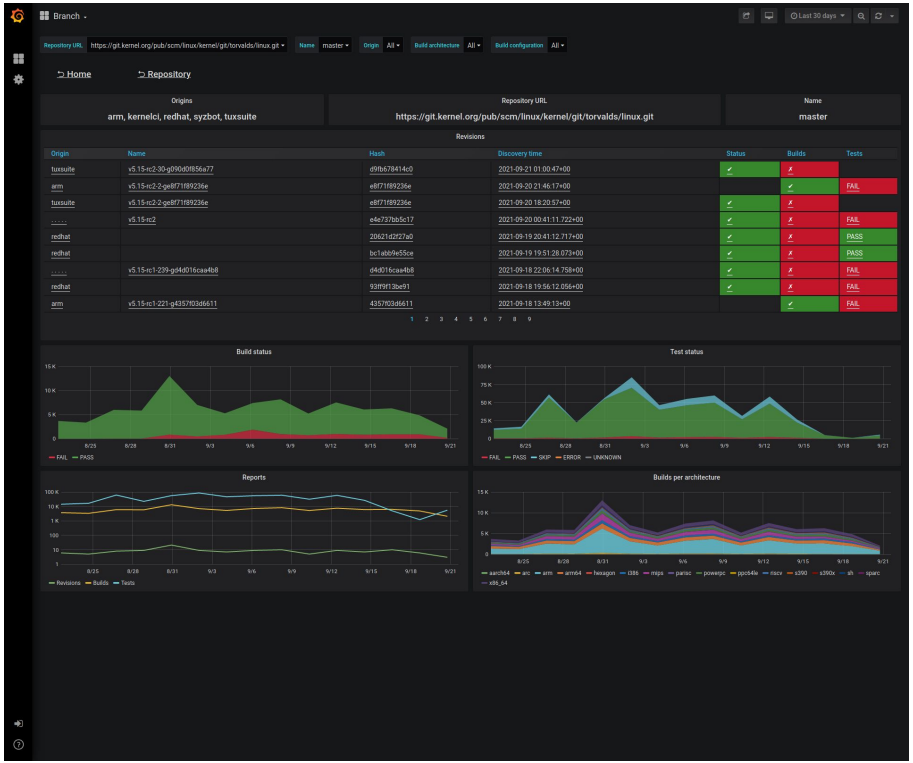


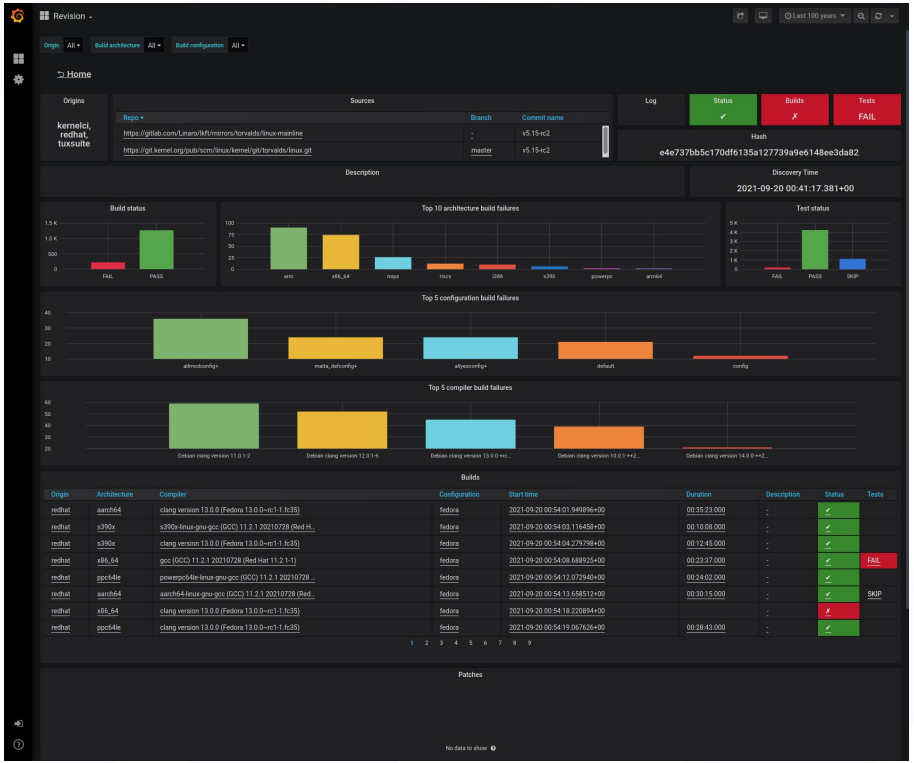
What are KernelCI and KCIDB





What are KernelCI and KCIDB





What are KernelCI and KCIDB

Below is the summary of results Kernel CI database has recorded
for this revision so far. See complete and up-to-date report at:

https://kcidb.kernelci.org/d/revision/revision?orgId=1&var-git_commit_hash=8008293888188c3923f5bd8a69370dae25ed14e5&var-patchset_hash=

OVERVIEW

Builds: ✗ FAIL
Tests: ✗ FAIL

REVISION

Status
✅ PASS
Commit
name: v5.16-rc7-108-g800829388818
hash: 8008293888188c3923f5bd8a69370dae25ed14e5
Checked out from
https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git master
https://gitlab.com/Linaro/lkft/mirrors/torvalds/linux-mainline
By
kernelci, redhat, syzbot, tuxsuite

BUILDS

Status
✗ 56 ✅ 738
Architectures

arm	✗ 26	✅ 347
riscv	✗ 8	✅ 21
mips	✗ 7	✅ 86
powerpc	✗ 5	✅ 47
x86_64	✗ 3	✅ 55
arm64	✗ 3	✅ 48
i386	✗ 2	✅ 42
s390	✗ 1	✅ 17
ppc64le	✗ 1	✅ 1
sh	—	✅ 24
arc	—	✅ 22
parisc	—	✅ 12
sparc	—	✅ 12
aarch64	—	✅ 2
s390x	—	✅ 2

Failures
✗ 4 riscv allnoconfig
✗ 2 arm allmodconfig "v5.16-rc7-108-g800829388818"
✗ 2 x86_64 allmodconfig "v5.16-rc7-108-g800829388818"
✗ 2 i386 allmodconfig "v5.16-rc7-108-g800829388818"
✗ 2 powerpc maple_defconfig
...
By
kernelci, redhat, syzbot, tuxsuite

TESTS

Status
✗ 23 ✅ 53 🚧✗ 5 🚧✅ 6 🚧⏭ 1
Failures
✗ baseline-qemu-docker
✗ baseline
✗ baseline-nfs
✗ ltp-mm
✗ cros-ec
...
By
kernelci, redhat, syzbot

See complete and up-to-date report at:

https://kcidb.kernelci.org/d/revision/revision?orgId=1&var-git_commit_hash=8008293888188c3923f5bd8a69370dae25ed14e5&var-patchset_hash=

LEGEND

🐞 ERROR	- Aborted. Test, tested code, or both might be faulty.
✗ FAIL	- Failed. Tested code is likely faulty.
✅ PASS	- Passed. Tested code is likely correct.
🆗 DONE	- Finished. Status of tested code is unknown.
⏭ SKIP	- Skipped. Planned, but didn't execute.
? UNKNOWN	- In progress, or status unknown.
🚧 WAIVED	- Waived result. Test is too new or shows known failures.
— BLANK	- No data, zero.

KernelCI

What's Continuous Integration?

KernelCI

What is CI?

1. Test every change

KernelCI

What is CI?

1. Test every change
2. Give feedback

KernelCI

The Metrics of CI

The metrics of CI

Coverage

How much functionality is tested

The metrics of CI

Coverage

How much functionality is tested

Latency

How fast feedback is produced

The metrics of CI

Coverage

How much functionality is tested

Latency

How fast feedback is produced

Reliability

How reliable feedback is

The metrics of CI

Coverage

How much functionality is tested

Latency

How fast feedback is produced

Reliability

How reliable feedback is

Accessibility

How easy it is to understand feedback

KernelCI The ✨ideal✨ CI

Coverage

Everything is tested

Latency

Instant feedback

Reliability

Always spot on

Accessibility

Just says what's broken

KernelCI The 💀worst💀 CI

Coverage

Useless

Latency

Takes forever

Reliability

Cannot be trusted

Accessibility

Incomprehensible

KernelCI The 💀worst💀 CI

Coverage

Useless

Latency

🎉 **Worse than NO CI** 🎉

Cannot be trusted

Accessibility

Incomprehensible

KernelCI

Where are we with upstream Kernel's CI?

Where are we?

- **Coverage**
  - Nobody seems to know

Coverage example

LCOV - code coverage report

				Hit	Total	Coverage
Current view:	top level		**Lines:**	425403	3375966	12.6 %
Test:	Coverage of kernel tests		**Functions:**	37119	220966	16.8 %
Date:	2023-06-07 00:04:15		**Branches:**	197204	2094174	9.4 %
Legend:	Rating:	low: < 25 % medium: >= 25 % high: >= 50 %				

Directory	Line Coverage		Functions		Branches	
arch/arm64/crypto	67.8 %	1075 / 1585	76.1 %	102 / 134	50.1 %	404 / 806
arch/arm64/hyperv	8.6 %	7 / 81	12.5 %	1 / 8	9.1 %	2 / 22
arch/arm64/include/asm	64.3 %	1379 / 2143	68.4 %	143 / 209	40.6 %	2837 / 6994
arch/arm64/include/asm/stacktrace	86.5 %	32 / 37	100.0 %	3 / 3	55.0 %	22 / 40
arch/arm64/include/asm/vdso	100.0 %	8 / 8	-	0 / 0	-	0 / 0
arch/arm64/kernel	44.5 %	3611 / 8106	51.1 %	435 / 852	27.5 %	1356 / 4923
arch/arm64/kernel/probes	5.2 %	20 / 383	6.0 %	3 / 50	2.0 %	5 / 247
arch/arm64/kvm	50.9 %	3097 / 6086	48.8 %	352 / 721	28.4 %	1155 / 4060
arch/arm64/kvm/hyp	33.0 %	462 / 1398	39.1 %	45 / 115	24.1 %	185 / 768
arch/arm64/kvm/hyp/include/hyp	64.5 %	247 / 383	81.2 %	26 / 32	23.8 %	76 / 320
arch/arm64/kvm/hyp/vhe	77.0 %	147 / 191	80.0 %	20 / 25	53.1 %	17 / 32
arch/arm64/kvm/vgic	42.2 %	1888 / 4471	52.1 %	189 / 363	25.7 %	615 / 2395
arch/arm64/lib	45.6 %	393 / 861	54.1 %	33 / 61	37.8 %	118 / 312
arch/arm64/mm	51.3 %	1079 / 2105	60.0 %	117 / 195	36.1 %	472 / 1306
arch/arm64/net	43.6 %	424 / 973	56.2 %	18 / 32	32.9 %	337 / 1025
block	63.7 %	11377 / 17858	66.8 %	1112 / 1664	47.2 %	5656 / 11975
block/partitions	57.1 %	463 / 811	70.8 %	34 / 48	37.7 %	203 / 539
certs	57.7 %	75 / 130	60.0 %	9 / 15	29.7 %	19 / 64

Where are we?

Coverage

Nobody seems to know

Where are we?

Coverage

Nobody seems to know

Latency

Several hours to weeks, pre-merge rare

Where are we?

Coverage

Nobody seems to know

Latency

Several hours to weeks, pre-merge rare

Reliability

Manual review required in many cases

Where are we?

Coverage

Nobody seems to know

Latency

Several hours to weeks, pre-merge rare

Reliability

Manual review required in many cases

Accessibility

Area-specific, disparate

KernelCI

Hard Limits

Hard Limits

Coverage

Hardware availability

Hard Limits

Coverage

Hardware availability

Latency

Hardware availability

Hard Limits

Coverage

Hardware availability

Latency

Hardware availability

Reliability

Hardware and kernel reliability

Hard Limits

Coverage

Hardware availability

Latency

Hardware availability

Reliability

Hardware and kernel reliability

Accessibility

Hardware availability, kernel complexity

KernelCI

Challenges

KernelCI Challenges

Coverage

Held back by other metrics

KernelCI Challenges

Coverage

Held back by other metrics

Latency

Unsafe pre-merge CI, slow human reviews

KernelCI Challenges

Coverage

Held back by other metrics

Latency

Unsafe pre-merge CI, slow human reviews

Reliability

Kernel/tests desync

KernelCI Challenges

Coverage

Held back by other metrics

Latency

Unsafe pre-merge CI, slow human reviews

Reliability

Kernel/tests desync

Accessibility

Disparate reports/dashboards

Challenges Compounded

Low **Reliability** and **Accessibility**

- Low developer trust towards results
  - No *gating*, constricted test/code feedback
    - Improvement process starved

Challenges Compounded

Low **Reliability** and **Accessibility**

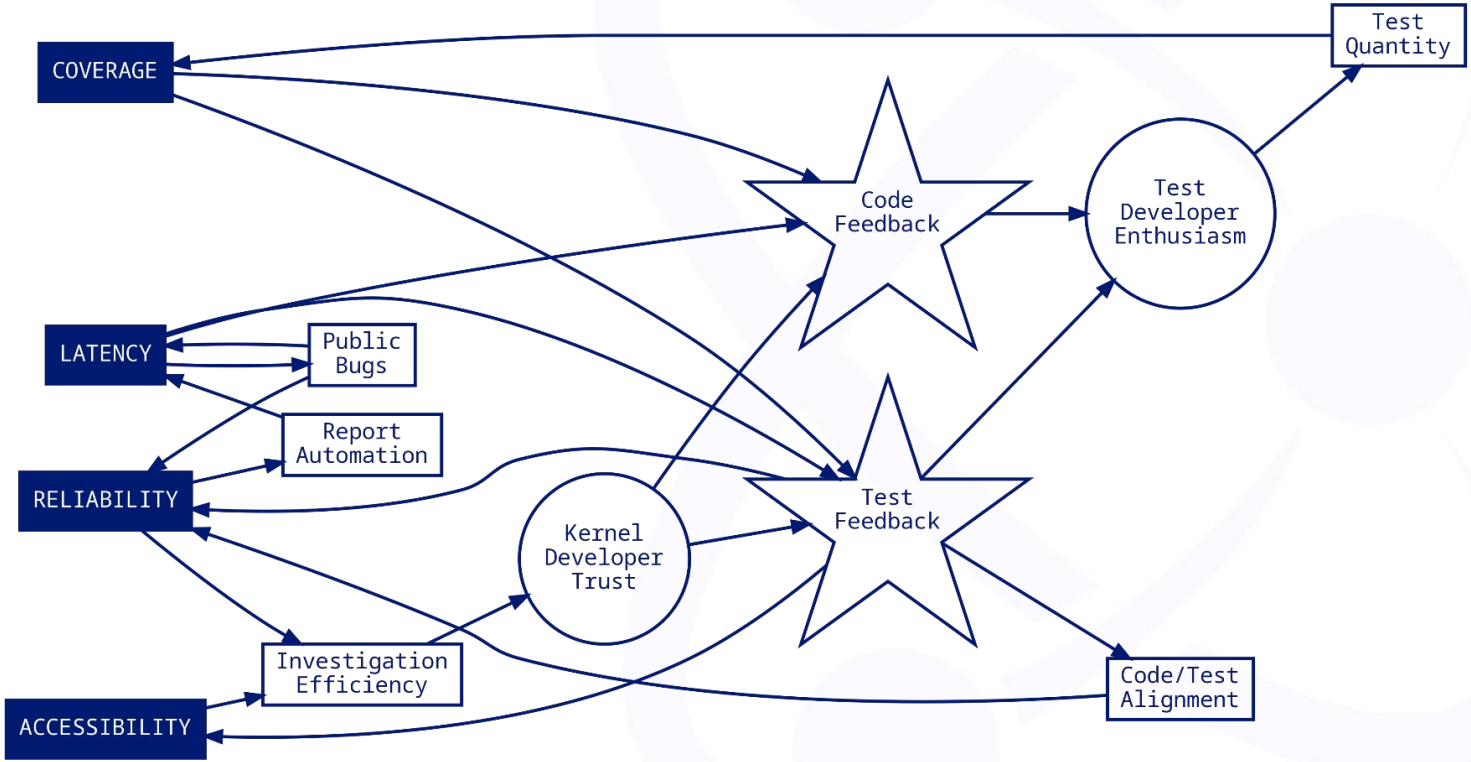
- Low developer trust towards results
  - No *gating*, constricted test/code feedback
    - Improvement process starved

High **Latency**

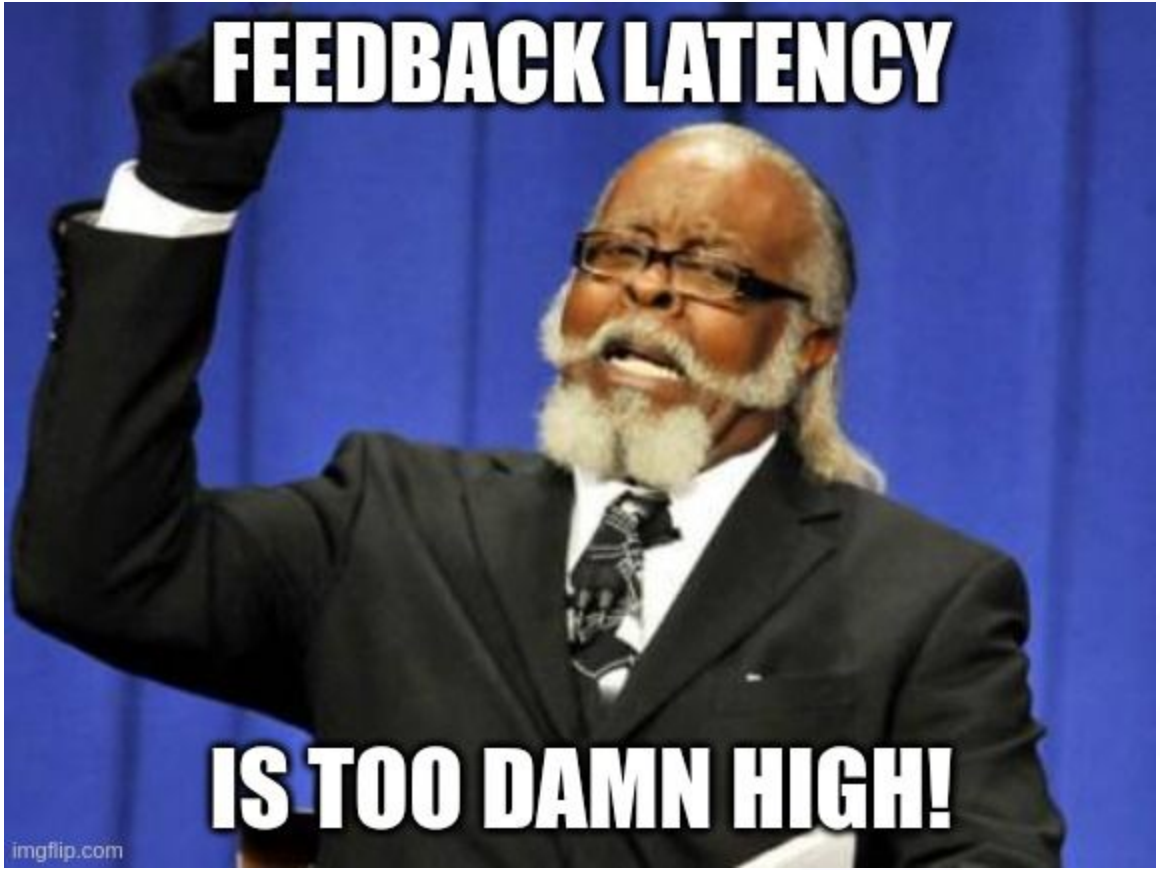
- No *gating*
- Bugs stay in public code longer
  - More time wasted by *everyone*
    - Less time for improvement
    - More **Latency**

KernelCI

All together now



All together now



KernelCI

Dog tax!

KernelCI

What can we do?

KernelCI

~~What can we do?~~

What we *can't* do?

What we *can't* do?

- Kernel community is not a single team
- Not a single company
- You **cannot force** people to look at test results
- "Enable gating then fix the tests" doesn't work
- We have to fix them first, gain ✨**developer trust**✨

KernelCI

The key thing

✨ Developer Trust ✨

KernelCI

So what *can* we do?

KernelCI Coverage

Attract more companies into testing
— more hardware, more tests, more coverage

- If you have your own CI system
  - KernelCI KCIDB interface is easy to use
  - Help improve KCIDB reporting
- If you'd like to contribute hardware
  - Set up a LAVA lab and connect to KernelCI
- Write to kernelci@lists.linux.dev

KernelCI Latency

More pre-merge testing — less bugs in public code, less time spent triaging, faster feedback!

- Use Patchwork for picking patches
  - Test safely what you can
- About 50 repos are already on GitHub/GitLab
  - Offer integration with your CI system
  - KernelCI is planning to offer its own
  - "The authenticated user gets the hardware"
- Encourage maintainers to use "git forges" for patches
  - CI integration and PR testing as selling points

KernelCI Latency

One test at a time

- Talk to maintainers
- Offer to test a staging branch
- Pick and choose a few most trusted tests to start
- KCIDB subscriptions can help with that
  - Want just that one test for your branch?
  - Particular compiler, architecture?
  - From all testers, or just some?
  - We're here for you - sign up!
    - Write to kernelci@lists.linux.dev

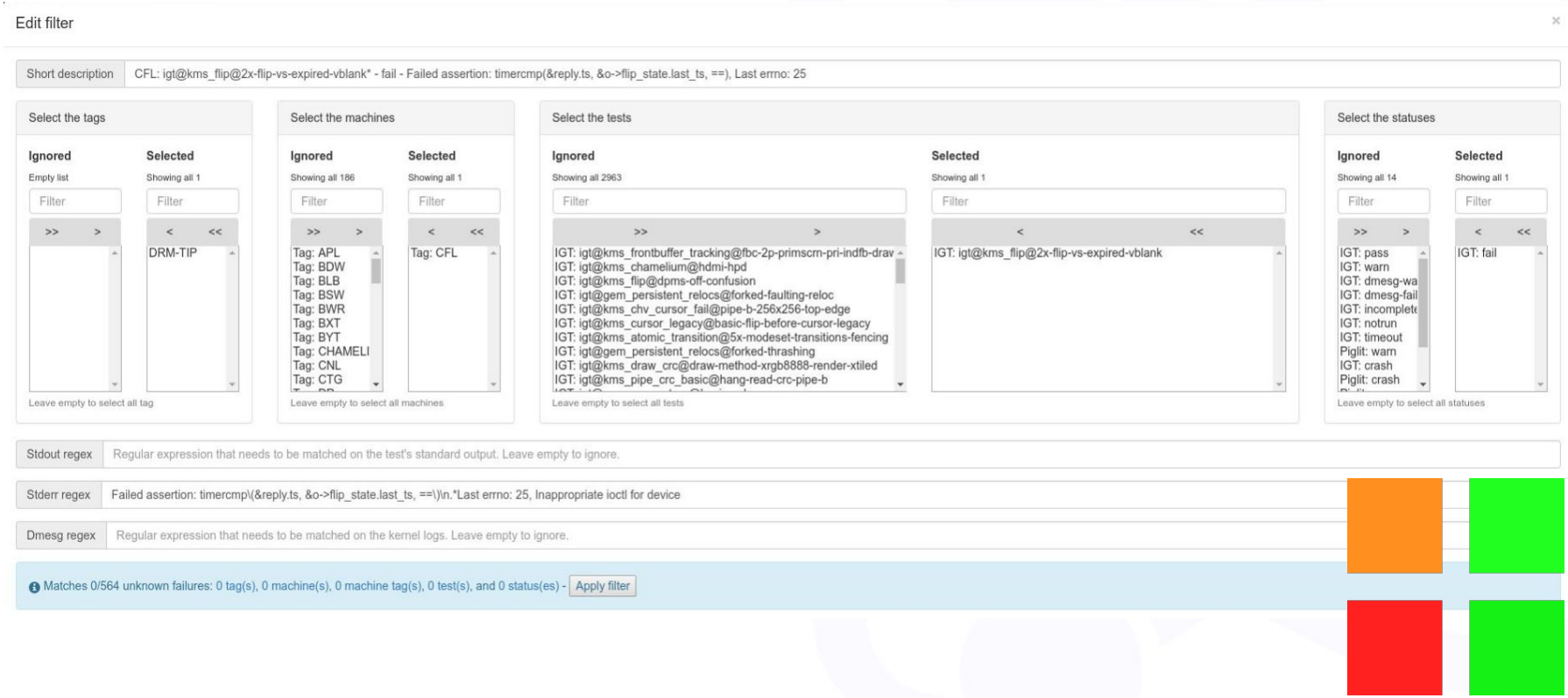
KernelCI

Latency

Automated triaging — more reliability, faster

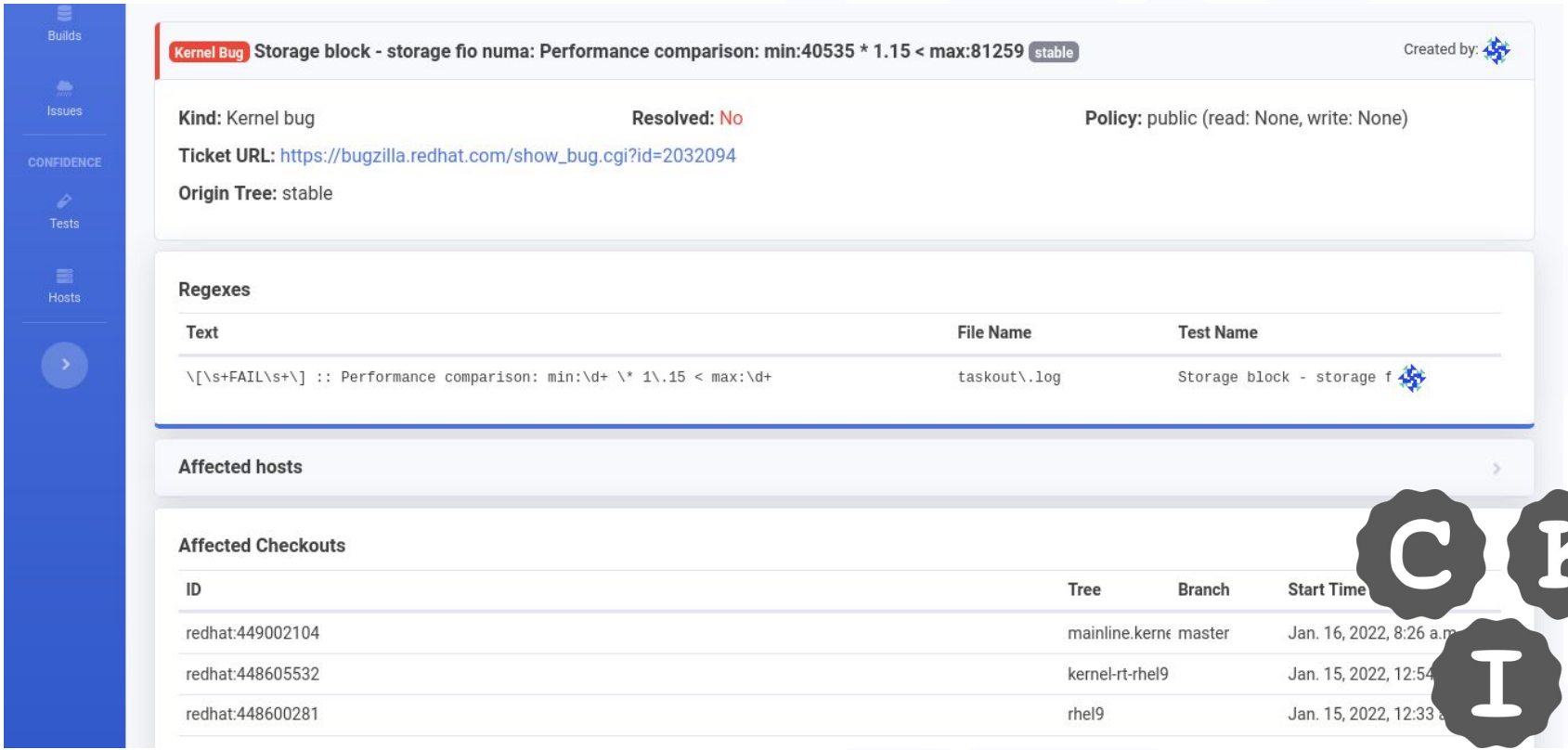
- Results still need reviews, let's make them easier
- KCIDB is building a triaging system for aggregated results, we need your help!
- References: GFX CI, CKI, syzbot

KernelCI Latency



https://archive.fosdem.org/2022/schedule/event/masking_known_issues_across_six_kernel_ci_systems/

KernelCI Latency



https://archive.fosdem.org/2022/schedule/event/masking_known_issues_across_six_kernel_ci_systems/

KernelCI Latency

syzbot Linux

sign-in | mailing list | source | docs

Open [1059] Fixed [3590] Invalid [7252]

KMSAN: uninit-value in sctp_inq_pop (2)
Status: upstream: reported C repro on 2022/01/08 08:00
Reported-by: syzbot+70a42f45e76bede082be@syzkaller.appspotmail.com
First crash: 58d, last: 1d21h

similar bugs (1):

Kernel	Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Patched	Status
upstream	KMSAN: uninit-value in sctp_inq_pop				3	204d	244d	0/22	auto-closed as invalid on 2021/10/

Sample crash report:

```
sctp_side_effects net/sctp/sm_sideeffect.c:1195 [inline]
sctp_do_sm+0x946f/0x9b50 net/sctp/sm_sideeffect.c:1166
sctp_assoc_bh_rcv+0xa15/0xdd0 net/sctp/associola.c:1054
sctp_inq_push+0x31c/0x440 net/sctp/inqueue.c:80
sctp_backlog_rcv+0x30f/0x10b0 net/sctp/input.c:344
sk_backlog_rcv include/net/sock.h:1030 [inline]
__release_sock+0x256/0x640 net/core/sock.c:2768
release_sock+0x98/0x2e0 net/core/sock.c:3300
sctp_wait_for_connect+0x52a/0x9e0 net/sctp/socket.c:9306
sctp_sendmsg_to_asoc+0x1c47/0x1f90 net/sctp/socket.c:1881
sctp_sendmsg+0x3eaa/0x5460 net/sctp/socket.c:2027
inet_sendmsg+0x15b/0x1d0 net/ipv4/af_inet.c:819
sock_sendmsg_nosec net/socket.c:704 [inline]
sock_sendmsg net/socket.c:724 [inline]
__sys_sendto+0x9ef/0xc70 net/socket.c:2036
__do_sys_sendto net/socket.c:2048 [inline]
__se_sys_sendto net/socket.c:2044 [inline]
__x64_sys_sendto+0x19c/0x210 net/socket.c:2044
do_syscall_x64 arch/x86/entry/common.c:51 [inline]
```

Google syzbot

KernelCI Latency

```
 sctp_sendmsg_to_asoc+0x1c47/0x1f90 net/sctp/socket.c:1881
 sctp_sendmsg+0x3eaa/0x5460 net/sctp/socket.c:2027
 inet_sendmsg+0x15b/0x1d0 net/ipv4/af_inet.c:819
 sock_sendmsg_nosec net/socket.c:704 [inline]
 sock_sendmsg net/socket.c:724 [inline]
 __sys_sendto+0x9ef/0xc70 net/socket.c:2036
 __do_sys_sendto net/socket.c:2048 [inline]
 __se_sys_sendto net/socket.c:2044 [inline]
 __x64_sys_sendto+0x19c/0x210 net/socket.c:2044
 do_syscall_x64 arch/x86/entry/common.c:51 [inline]
 do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82
 entry_SYSCALL_64_after_hwframe+0x44/0xae

CPU: 0 PID: 3479 Comm: syz-executor678 Not tainted 5.16.0-rc5-syzkaller #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011
==================================================
```

Crashes (12):

Manager	Time	Kernel	Commit	Syzkaller	Config	Log	Report	Syz repro	C repro	VM info	
ci-upstream-kmsan-gce	2022/01/04 02:03	https://gith...	81c325bbf94e	4a3f34f2	.config	log	report	syz	C		KMSAN: unin
ci-upstream-kmsan-gce	2022/01/09 14:48	https://gith...	81c325bbf94e	2ca0d385	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/09 03:03	https://gith...	81c325bbf94e	2ca0d385	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/08 23:49	https://gith...	81c325bbf94e	2ca0d385	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/08 08:29	https://gith...	81c325bbf94e	2ca0d385	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/07 15:49	https://gith...	81c325bbf94e	2ca0d385	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/07 07:08	https://gith...	81c325bbf94e	6acc789a	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce	2022/01/03 22:19	https://gith...	81c325bbf94e	4a3f34f2	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce-386	2022/01/14 20:25	https://gith...	fa3879a274df	53e00b45	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce-386	2021/12/03 12:30	https://gith...	093998ececa3	c7c20675	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce-386	2021/11/26 04:17	https://gith...	a535b0caaa2f	63eeac02	.config	log	report			info	KMSAN: unin
ci-upstream-kmsan-gce-386	2021/11/19 07:31	https://gith...	412af9cd936d	31a30fc0	.config	log	report			info	KMSAN: un

Google syzbot

Reliability

Move tests in-tree — no desync with code, "official" tests attract contributions

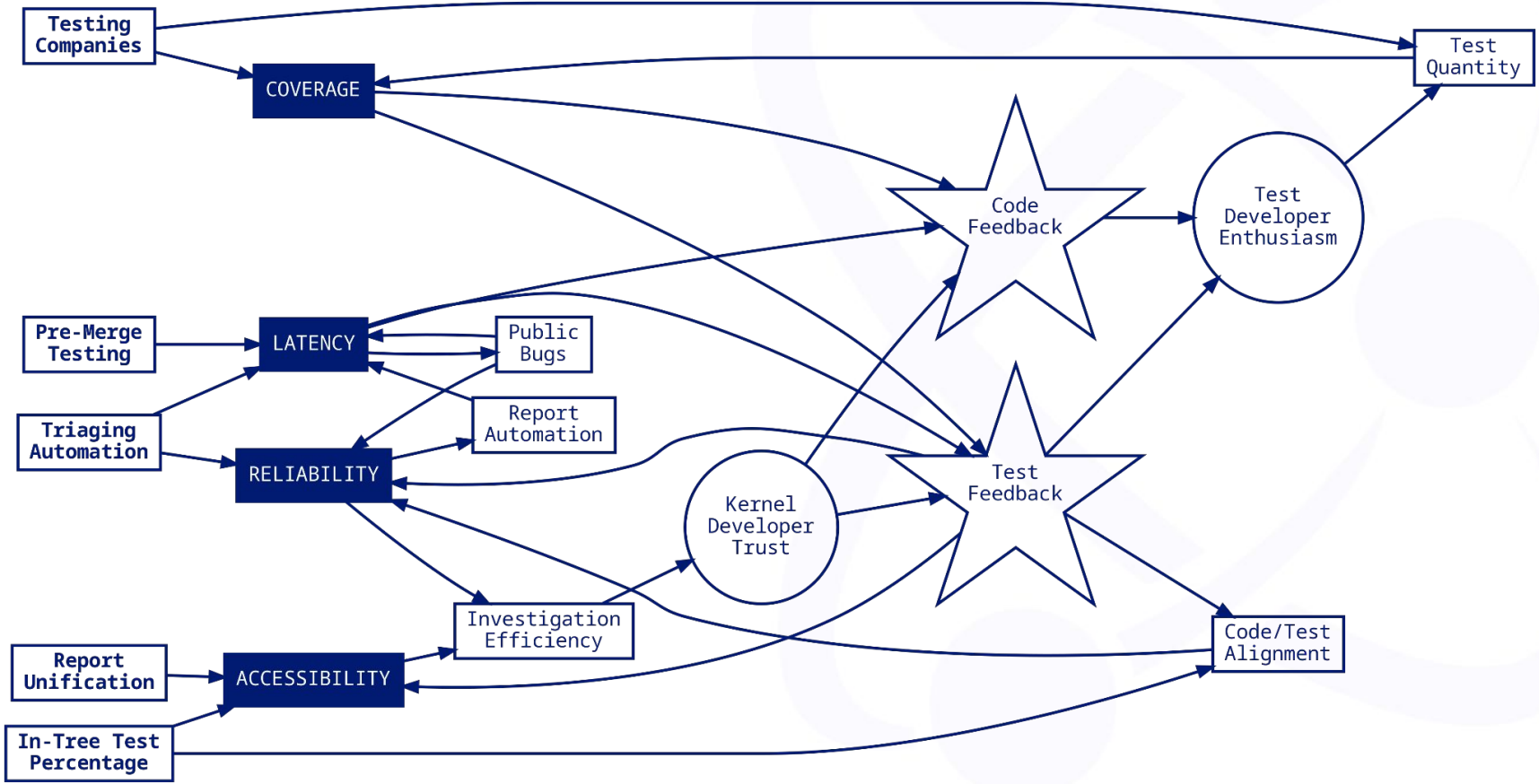
- Start with most popular and well-developed ones
  - LTP?
- Integrate into kernel documentation
- Help backport to older branches
  - Stable?
- Prioritize execution of in-tree tests in CI
  - Shorten the loop

Accessibility

Uniform reports/dashboards — time saved

- We're here for you - sign up!
  - Write to kernelci@lists.linux.dev
- Give us your requirements and use cases!
- Contribute your reporting templates!
- Help development!

All together now



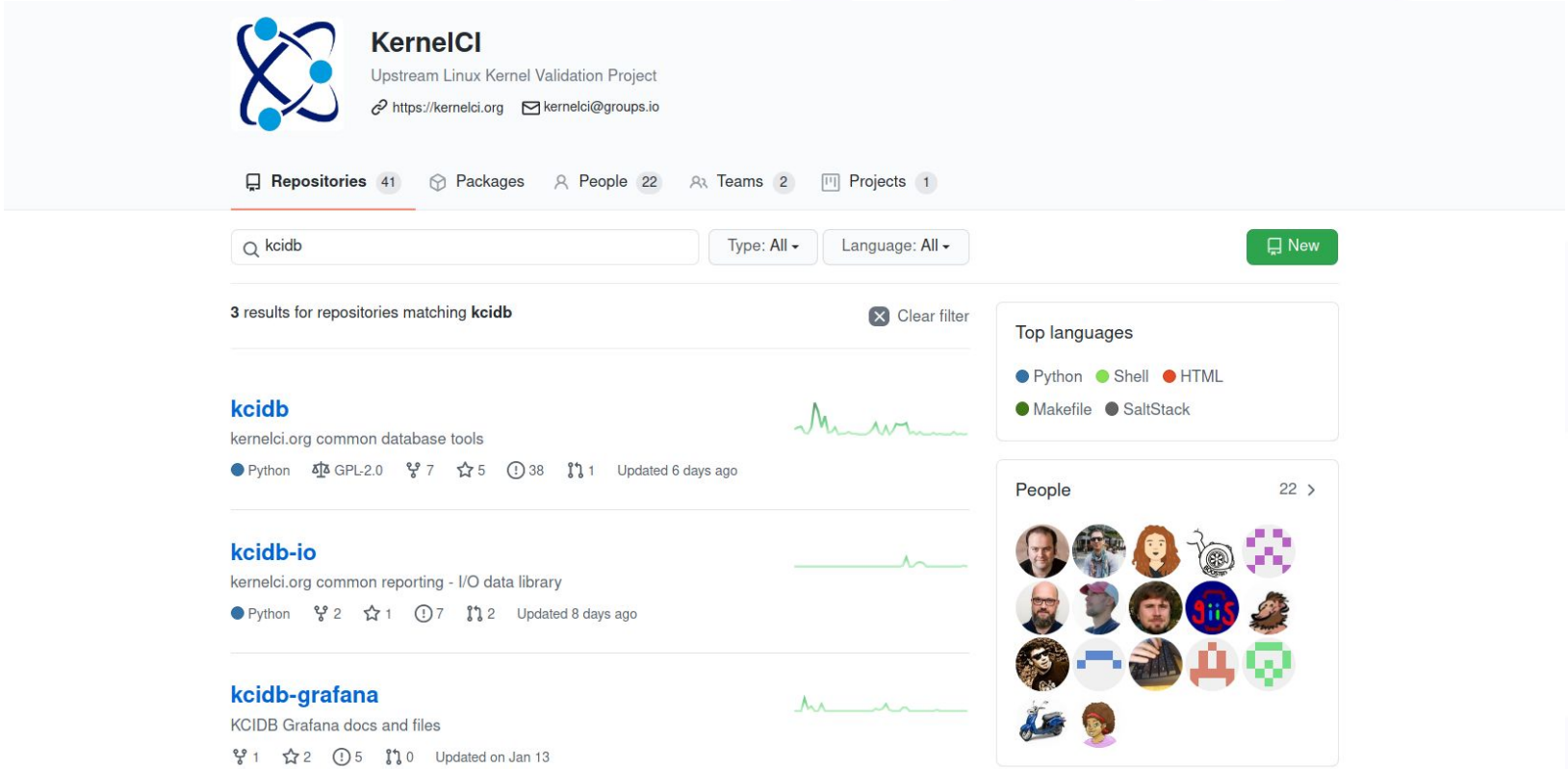
KernelCI

Thank you!

KernelCI

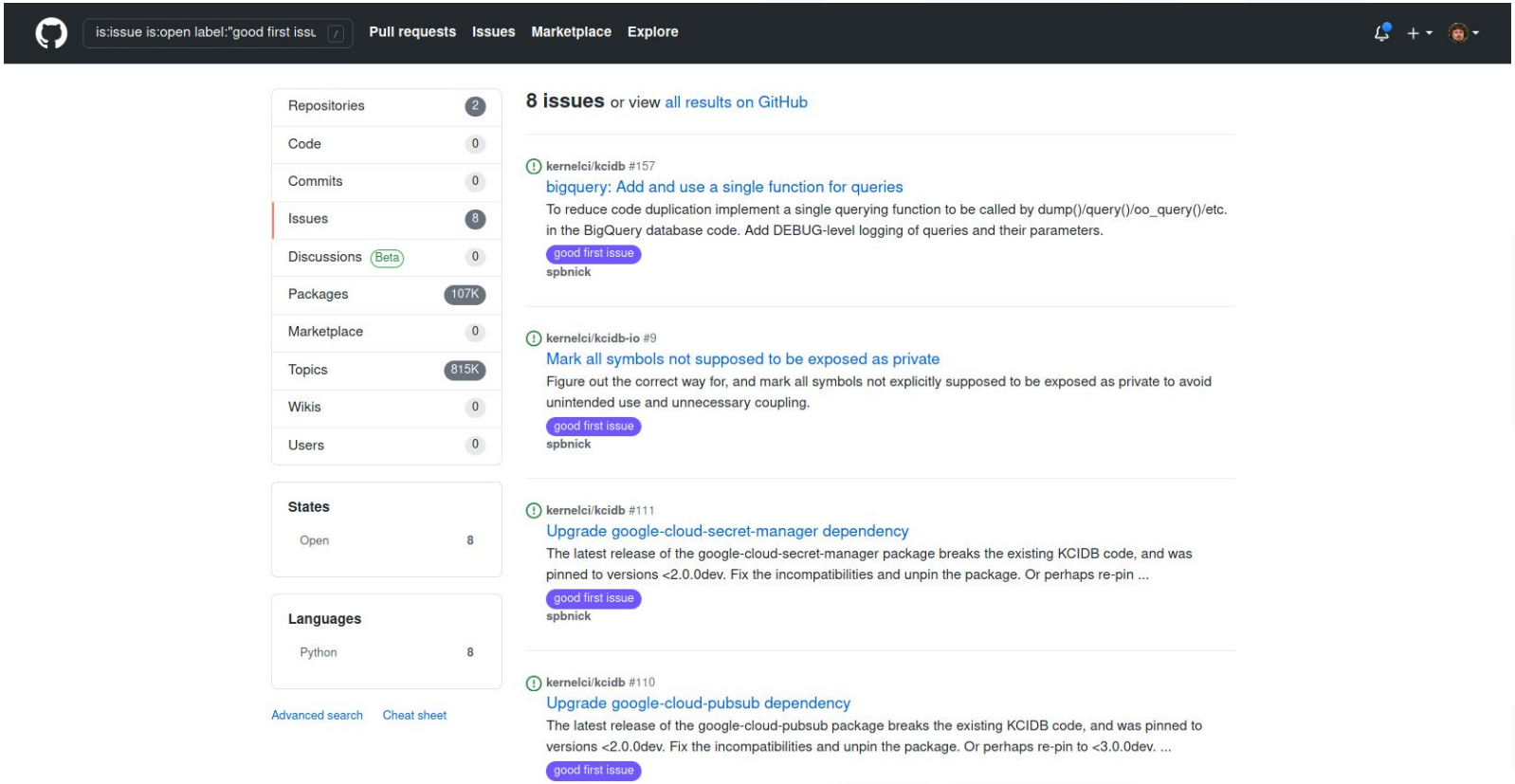
Join Us!

Main repos on GitHub

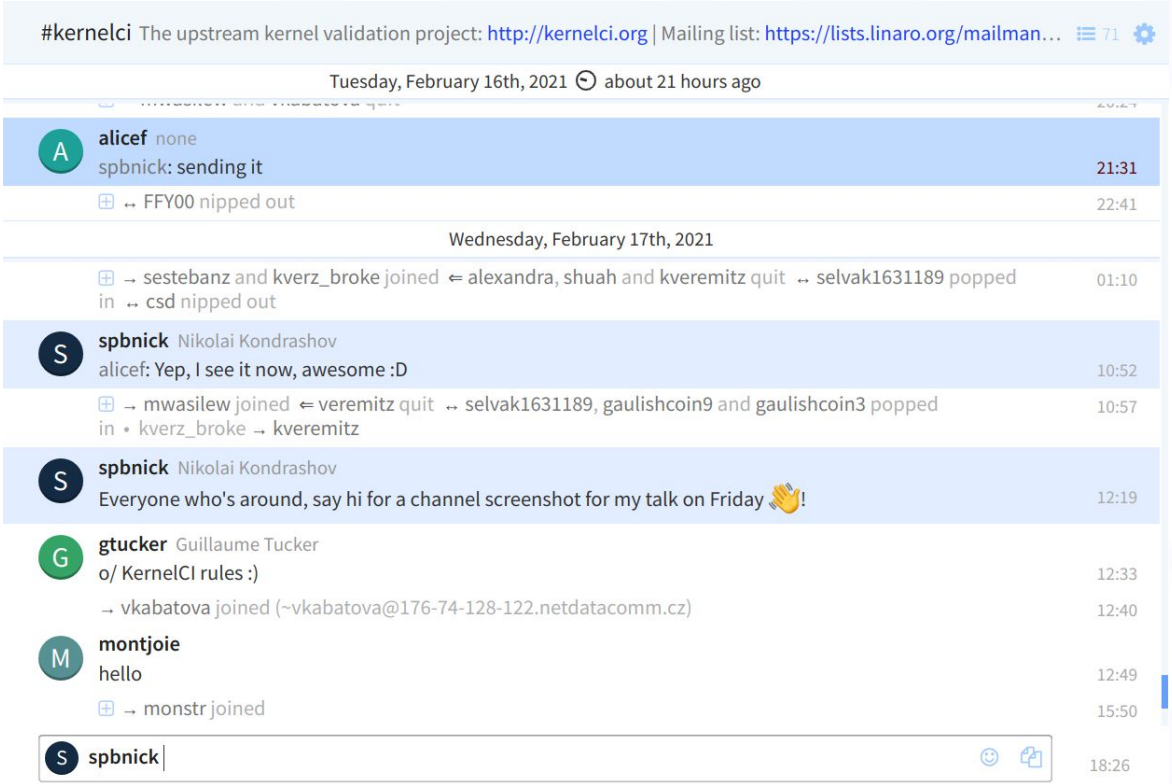


https://github.com/kernelci/?q=kcidb

A bunch of "good first issues"



https://github.com/search?q=is:issue...

IRC channel

#kernelci at libera.chat