Test-driven kernel releases

Guillaume Tucker
gtucker@collabora.com
Development & Testing

- Kernel development
- Manual testing
- Automated testing
Automated Testing
Open Source Philosophy

- Single mainline code base
- Many contributors
- Many use-cases
- Application changes are sent upstream
- Reduced duplication of efforts
Open Testing Philosophy

- Single mainline code base including tests
- Many contributors who run tests
- Test results sent upstream
- Test results summary in each release
- Reduced duplication of testing efforts
Hidden Mass of Testing

Duplicated testing efforts

No solution for tracking results upstream

Testing stays hidden as if it was downstream
automated testing
syzbot

https://syzkaller.appspot.com/

syzcall fuzzing
Automated bisection
Reproducers
Web UI
KernelCI

https://linux.kernelci.org/job/

Tailored CI system
Web API
Distributed test labs
Kubernetes
Automated bisection
KCIDB database

The results shown here cover the last 14 days of available data starting from Fri, 25 Mar 2022 (time is UTC based).

<table>
<thead>
<tr>
<th>Tree</th>
<th>Branch</th>
<th>Latest Build Status</th>
<th>Latest Test Results</th>
<th>Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>stable-rc</td>
<td>linux-4.14.y</td>
<td>44 3 2</td>
<td>0 0 0</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>android</td>
<td>android12-5.4</td>
<td>93 45 4</td>
<td>24 2 0</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>soc</td>
<td>for-next</td>
<td>122 4 3</td>
<td>939 16 2</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>android</td>
<td>android-mainline</td>
<td>109 29 3</td>
<td>1,352 11 2</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>soc</td>
<td>arm/fixes</td>
<td>123 3 2</td>
<td>2,647 137 16</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>android</td>
<td>android11-5.4</td>
<td>87 75 12</td>
<td>485 21 12</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/4.14</td>
<td>119 6 1</td>
<td>33 3 0</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>clk</td>
<td>clk-next</td>
<td>130 4 1</td>
<td>1,551 32 6</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/5.10</td>
<td>129 2 3</td>
<td>1,391 129 12</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>next</td>
<td>master</td>
<td>146 9 4</td>
<td>3,429 211 54</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/5.16</td>
<td>107 2 2</td>
<td>1,069 53 9</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>net-next</td>
<td>master</td>
<td>127 5 3</td>
<td>2,861 136 28</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>mainline</td>
<td>master</td>
<td>141 5 7</td>
<td>3,608 140 26</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/5.4</td>
<td>115 9 1</td>
<td>1,044 81 8</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>next</td>
<td>pending-fixes</td>
<td>131 2 2</td>
<td>4,844 190 67</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/4.19</td>
<td>139 10 4</td>
<td>1,136 118 9</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
<tr>
<td>stable-rc</td>
<td>queue/5.15</td>
<td>125 1 3</td>
<td>1,059 33 19</td>
<td>2022-03-25</td>
<td>🔴</td>
</tr>
</tbody>
</table>
Red Hat CKI

https://datawarehouse.cki-project.org/

Fedora kernels

Mainline kernels

Stable

LTP, kselftest etc.

KCIDB integration
Linux kernel regression status

[prev] [next] [mainline] [stable/longterm] [dormant] [resolved] | [new] | [all]

current cycle (v5.18..aka v5.18-post), culprit identified
none known by regzbot

---

previous cycle (v5.17..v5.18), culprit identified, with activity in the past three months

- **b04314aeb01**
  (v5.18-r1c1)
  ![mm: chiq_test runs 7 minutes instead of ~1 second](https://github.com/linaro-regtools/regzbot/pull/206)
  Earliest & latest: 9 days ago. Noteworthy: [patch].

- **b06e6fa4611**
  (v5.18-r1c1)
  ![mm: [mm/page_alloc] f26240acle: perf/throughput: -18.0%](https://github.com/linaro-regtools/regzbot/pull/211)
  Earliest & latest: 10 days ago. Noteworthy: [patch].

---

older cycles (v5.17), culprit identified, with activity in the past three months

- **453c41005183**
  (v5.12-r1c1)
  ![boot suspend broken on Lenovo T460p](https://github.com/linaro-regtools/regzbot/pull/212)
  Earliest & latest: 3 days ago. Noteworthy: [1].

- **4c5c7208b376**
  (v5.12-r1c1)
  ![qla2xxx: tape drive not removed after unplug FC cable](https://github.com/linaro-regtools/regzbot/pull/213)
  Earliest & latest: 3 days ago.

- **e5910bc63c59**
  (v5.17-r1c1)
  ![SEGY in Xen PV doesn't after updating from 5.16.10 to 5.17.5](https://github.com/linaro-regtools/regzbot/pull/214)
  Earliest & latest: 3 days ago.

- **a5a31138cb8d**
  (v5.17-r1c1)
  ![BUG BLE device unpairing triggers kernel panic](https://github.com/linaro-regtools/regzbot/pull/215)
  Earliest & latest: 3 days ago. Noteworthy: [patch].

- **5d06b79d917**
  (v5.17-r1c1)
  ![pci: bcm43854: CM4 no longer boots from SD card](https://github.com/linaro-regtools/regzbot/pull/216)
  Earliest & latest: 3 days ago. Noteworthy: [1], [2]

- **f090a00e2c09**
  (v5.11-r1c1)
  ![memory corruption on Amel SAMASDI](https://github.com/linaro-regtools/regzbot/pull/217)
  Earliest & latest: 3 days ago. Noteworthy: [patch].

- **a199c3e46a9**
  (v5.16-r1c1)
  ![scsi: pm: Resume process hangs for 5-6 seconds starting sometime in 5.16](https://github.com/linaro-regtools/regzbot/pull/218)
  Earliest & latest: 3 days ago. Noteworthy: [1].

---

previous cycle (v5.17..v5.18), unknown culprit, with activity in the past three weeks
none known by regzbot

---

older cycles (v5.17), unknown culprit, with activity in the past three weeks

- **v5.16..v5.17**
  ![fsocache: file contents are corrupted](https://github.com/linaro-regtools/regzbot/pull/219)
  Earliest & latest: 4 days ago. Noteworthy: [patch].

- **v5.10..v5.15**
  ![netfilter: interleaved performance regression related to ipset](https://github.com/linaro-regtools/regzbot/pull/220)
  Earliest & latest: 7 days ago.

- **v5.12..v5.13**
  ![NFSv4 stops serving clients](https://github.com/linaro-regtools/regzbot/pull/221)
  Earliest & latest: 10 days ago. Noteworthy: [1]

- **v5.16..v5.17**
  ![Hans in 5.17-4.6 that appears to be due to xen](https://github.com/linaro-regtools/regzbot/pull/222)
  Earliest & latest: 10 days ago. Noteworthy: [2].

---

All known regressions

Essentially manual submissions

Seamless integration with emails

Weekly report on LKML for mainline

---

regzbot

https://linux-regtracking.leemhuis.info/regzbot/mainline/
show me the results
Focusing on the results

Manual runs

Maintainer scripts

Automated systems
Focusing on the results

Manual runs
Maintainer scripts
Automated systems

Results are the least common denominator
Benefits of results in releases

Valuable for users in general

Canonical way to keep track of code quality

Essentially, avoiding the “works for me” syndrome

[10:45:37] Testing complete. 115 tests run. 0 failed. 0 crashed.
[10:45:37] Test results stored in /root/linux/kunit.json
[10:45:37] Elapsed time: 265.516s total, 0.007s configuring, 41.982s building, 0.000s running
Challenges

Shift in workflow: results are needed before the release

- Similar to how -rc works for stable and mainline
- Expect positive results rather than solely look for regressions
- Additional step for maintainers

Keeping it simple and not disruptive

- Optional
- Up to each maintainer to decide which results to include
in practice
Where to start?

Results reproducible on any hardware

Tests included in the kernel source tree

- Plain builds with reference toolchain binaries, Docker images
- Builds with sparse enabled: `make C=1`
- coccicheck
- KUnit
- Device tree validation
Where to start?

Results reproducible on any hardware

Tests included in the kernel source tree

- Plain builds with reference toolchain binaries, Docker images
  - Builds with sparse enabled `make C=1`
- cocci
cHECK
- KUnit

Device tree validation

Documentation: https://docs.kernel.org/
RFC 1: Test results in-tree

Similar to linux-next merge logs

Updated for each release (stable, mainline, -next)

Rely on Git history for older results

Results
├── kselftest
│   ├── futex
│   └── lkdtm
└── KUnit
    └── results.json
        summary
RFC 2: Test-link in commit

commit b843adea138649361a8d68b4c9d5824c33fb0146
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date: Wed May 25 09:18:02 2022 +0200

Linux 5.10.118

Link: https://lore.kernel.org/r/20220523165812.244140613@linuxfoundation.org
Test-link: https://testing.kernel.org/stable/5.10.118
Tested-by: Florian Fainelli <f.fainelli@gmail.com>
Tested-by: Shuah Khan <skhan@linuxfoundation.org>
Tested-by: Fox Chen <foxhlchen@gmail.com>
Tested-by: Salvatore Bonaccorso <carnil@debian.org>
Tested-by: Sudip Mukherjee <sudip.mukherjee@codethink.co.uk>
Tested-by: Pavel Machek (CIP) <pavel@denx.de>
Tested-by: Hulk Robot <hulkrobot@huawei.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
RFC 3: Git meta-data

Tied to Git history

Separate from commit merge workflow

Similar to Git notes

```
git results show REVISION
```
Some thoughts

Subsystem-specific results in separate location?
  Integration results for mainline / stable / linux-next
  Subsystem results could be pulled in alongside code

Follow regular email workflow for adding results
  Keep in-tree result summaries in plain text
  Extra data can be hosted on separate systems
RFC: <your idea here>

How does the concept sound?

Has this been tried or discussed before?

Does it seem worth the effort?

Time for an RFC on LKML to go through some details?
Thank you!