Intel GFX CI

Doing validation the Linux Way

Martin Peres - Intel's Open Source Graphics Center

Feb 2nd 2019

Agenda

- Linux's unique development model
- How to prevent regressions from getting in?
- Case study: Intel GFX CI
- Conclusion

Linux and its unique development model

- The Linux kernel is massive:
  - 1000s of drivers in one tree and 10000+ configuration parameters
  - 1600+ developers, 10+% of hobbyists and 250 companies contribute each release (Intel #1)
  - ~17M lines of code across 50k files
  - 100s of integration trees and 5 stable trees
  - 63 to 70 days between releases
  - ~14k commits per release
  - 7.8 commits per hour in average in the main tree

Linux and its unique development model

- The Linux kernel has no architects, but it has rules:
  - No user-visible regression: if updating breaks a program, the change is reverted.
  - No new kernel feature without an open source userspace (especially true for DRM).
- These rules made Linux go from a niche Operating System, to the most used one:
  - Strictly-improving Software means each new contribution increases the user base
- However, in practice, regressions do come in:
  - This is why your phone is still running prehistoric kernels
  - This dilutes the development of Linux, and is equivalent to forking it

How to prevent regressions?

Why do regressions get in?

- Upstream Linux is a validation nightmare:
  - Single code-base, with high-level of code sharing between drivers
  - One version every 2-3 months
  - Developers typically can only test their code on one machine
  - General lack of test suites ready for automated-testing
  - Few unit tests (although there is a project for this)
  - Few kernel self tests (fewer than 1000)

- Traditional human-powered QA falls short:
  - Too many HW/SW configurations, use cases, and unwritten expectations
  - By the time a test cycle is done, the tree is already outdated
  - Instead, Linux relies on user-testing during -rc cycles, but few users test these

Why do we need Continuous Integration (CI)?

- Pre-merge testing allows putting the cost of integration on the person making changes:
  - less time spent on bug fixing in post merge (where reverts are hard to get accepted);
  - provides better global understanding to developers;
  - keeps the integration tree in working condition at all time;
  - it scales better with the number of developers!
- Challenges:
  - The test system needs to be fast, so as patches don't get merged before being tested
  - The test system needs to run public tests which are ready for automated testing
  - Keeping the integration tree working is difficult:
    - back merges from Linux bring thousands of line of code without integration testing.
  - Filtering known issues to provide curated pre-merge testing reports

Providing useful pre-merge reports to developers

- Provide all the necessary information to understand failures:
  - Machine information (dmidecode, kernel logs, connected displays, …)
  - Full logs of the test execution (stdout, stderr, dmesg)
  - Push each tested version of a component as a tag in a public repo
  - Store the compiled versions of each components

- Concentrate on what the developer changed:
  - Integration testing is extremely noisy (especially when involving boot and suspend)
  - Known issues need to be labeled and/or filtered out
  - Show the list of components that changed

How to filter known issues?

- We need a tool allowing:
  - Post-merge issues' signatures/filters to be created automatically or manually
  - Signatures/Filters need to be associated to bugs tracking them
  - Filtered pre-merge reports to use the signatures to filter out the known issues
  - Developers to prioritize fixing issues based on their impact
  - Bonus: trigger an auto-bisection using the CI idle time of machines

- Such a tool is not a utopia:
  - CI Bug Log was created with these goals in mind one year ago
  - Led to myself filing over 700 bugs last year, and reducing the pre-merge noise level
  - Open sourced a week ago: https://gitlab.freedesktop.org/gfx-ci/cibuglog

CI Bug Log: Example of a report

CI Bug Log - changes from CI_DRM_5488 -> Patchwork_12046
==========

SUCCESS
No regressions found.
External URL: https://patchwork.freedesktop.org/api/1.0/series/55750/re...

Known issues

Here are the changes found in Patchwork_12046 that come from known issues:

IGT changes
Issues hit
* igt@gem_exec_suspend@basic-s4-devices:
  - fi-blb-e6850: PASS -> INCOMPLETE [fdo#107718]
* igt@kms_chamelium@hdmi-hpd-fast:
  - fi-kbl-7500u: PASS -> FAIL [fdo#108767]

Possible fixes
* igt@kms_chamelium@dp-edid-read:
  - fi-kbl-7500u: WARN -> PASS
* igt@kms_pipe_crc_basic@read-crc-pipe-b-frame-sequence:
  - fi-byt-clapper: FAIL [fdo#103191] / [fdo#107362] -> PASS +1

[fdo#103191]: https://bugs.freedesktop.org/show_bug.cgi?id=103191
[fdo#107362]: https://bugs.freedesktop.org/show_bug.cgi?id=107362
[fdo#107718]: https://bugs.freedesktop.org/show_bug.cgi?id=107718
[fdo#108767]: https://bugs.freedesktop.org/show_bug.cgi?id=108767

Participating hosts (44 -> 40)

Missing (4): fi-kbl-soraka fi-ilk-m540 fi-byt-squawks fi-bsw-cyan

Build changes
---
* Linux: CI_DRM_5488 -> Patchwork_12046

CI_DRM_5488: f13eede6ea3e780d900c5220bf09d764a80a3a8f @ git://anongit.freedesktop.org/gfx-ci/linux
IGT_4790: dcdf4b04e16312f8f52ad389388d834f9d74b8f0 @ git://anongit.freedesktop.org/xorg/app/intel-gpu-tools
Patchwork_12046: 6f40b811103eee129743c6465e987be7a51e7596 @ git://anongit.freedesktop.org/gfx-ci/linux

== Linux commits ==
6f40b811103e drm/i915/execlists: Suppress redundant preemption
2ee9b7413598 drm/i915/execlists: Suppress preempting self
0cf0a44086c4 drm/i915: Rename execlists->queue_priority to preempt_priority_hint

CI Bug Log: Example of a filter

Edit filter

Short description: CFL: igt@kms_flip@2x-flip-vs-expired-vblank* - fail - Failed assertion: timercmp(&reply.ts, &o->flip_state.last_ts, ==), Last errno: 25

Select the tags

Ignored	Selected
Empty list	Showing all 1
Filter	Filter
>> >	< <<
	DRM-TIP

Leave empty to select all tag

Select the machines

Ignored	Selected
Showing all 186	Showing all 1
Filter	Filter
>> >	< <<
Tag: APL Tag: BDW Tag: BLB Tag: BSW Tag: BWR Tag: BXT Tag: BYT Tag: CHAMELI Tag: CNL Tag: CTG	Tag: CFL

Leave empty to select all machines

Select the tests

Ignored	Selected
Showing all 2963	Showing all 1
Filter	Filter
>> >	< <<
IGT: igt@kms_frontbuffer_tracking@fbc-2p-primscrn-pri-indfb-drav IGT: igt@kms_chamelium@hdmi-hpd IGT: igt@kms_flip@dpms-off-confusion IGT: igt@gem_persistent_relocs@forked-faulting-reloc IGT: igt@kms_chv_cursor_fail@pipe-b-256x256-top-edge IGT: igt@kms_cursor_legacy@basic-flip-before-cursor-legacy IGT: igt@kms_atomic_transition@5x-modeset-transitions-fencing IGT: igt@gem_persistent_relocs@forked-thrashing IGT: igt@kms_draw_crc@draw-method-xrgb8888-render-xtiled IGT: igt@kms_pipe_crc_basic@hang-read-crc-pipe-b	IGT: igt@kms_flip@2x-flip-vs-expired-vblank

Leave empty to select all tests

Select the statuses

Ignored	Selected
Showing all 14	Showing all 1
Filter	Filter
>> >	< <<
IGT: pass IGT: warn IGT: dmesg-wa IGT: dmesg-fail IGT: incomplete IGT: notrun IGT: timeout Piglit: warn IGT: crash Piglit: crash	IGT: fail

Leave empty to select all statuses

Stdout regex: Regular expression that needs to be matched on the test's standard output. Leave empty to ignore.

Stderr regex: Failed assertion: timercmp\(&reply.ts, &o->flip_state.last_ts, ==\)\n.*Last errno: 25, Inappropriate ioctl for device

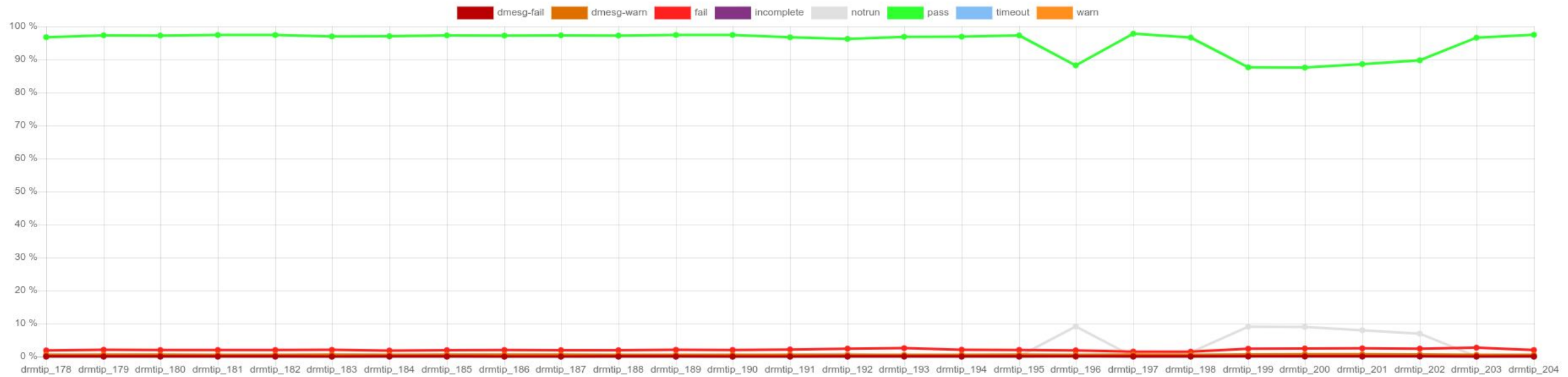
Dmesg regex: Regular expression that needs to be matched on the kernel logs. Leave empty to ignore.

Matches 0/564 unknown failures: 0 tag(s), 0 machine(s), 0 machine tag(s), 0 test(s), and 0 status(es) - Apply filter

Close | Check | Edit

CI Bug Log: Most hitting bugs

Passrate trend



Tabular view of the passrate (27 runconfigs, 8 statuses)

Most-hit issues (81)

Bug ID ▲▼	Bug Status ▲▼	Last Updated ▲▼	Bug Summary ▲▼	Hit rate ▲▼
• fdo#108145	• CLOSED/WONTFIX	• 115 days, 0:51:45 ago	• [CI][SHARDS] igt@kms_plane_alpha_blend@* - fail - Failed assertion: !mismatch (gen8-10)	0.73% (2774 / 379938)
• fdo#107956	• NEW	• 133 days, 2:08:54 ago	• [CI][SHARDS] igt@kms_busy@extended_*_render-[abc] - dmesg-warn - Asynchronous wait on fence i915:kms_busy\[\d+\]/0:1 timed out \(hint:intel_atomic_commit_ready	0.43% (1648 / 379938)

CI Bug Log: Open bugs needing attention

Overdue Bugs (420 / 642)

Deadlining Bugs (45 / 642)

Bug ID ▲▼	Status ▲▼	Summary ▲▼	Component ▲▼	Features ▲▼	Platforms ▲▼	Assignee ▲▼	Priority ▲▼	Involving (dev, user) ▲▼	Created ▲▼	Last Updated by user ▲▼	Last Updated by dev ▲▼	SLA (days) ▲▼	SLA deadline ▲▼
fdo#108546	ASSIGNED	Loading i915 kernel module breaks NVMe PCI device on the new Coffee Lake box	DRM/Intel	No features selected	CFL	*Intel GFX Bugs mailing list <intel-gfx-bugs@lists.freedesktop.org>*	medium	1. *Takashi Iwai <tiwai@suse.de>* (17) 2. Rodrigo Vivi <rodrigo.vivi@gmail.com> (6) 3. Lakshmi <lakshminarayana.vudum@intel.com> (2) 4. Jani Nikula <jani.nikula@intel.com> (1) 5. Ville Syrjala <ville.syrjala@linux.intel.com> (1)	Oct. 25, 2018, 6:44 a.m.	66 days, 2:22:08 ago	59 days, 16:11:53 ago	60	7:48:07
fdo#103167	REOPENED	[CI] igt@kms_frontbuffer_tracking@* - fail - CRC mismatch	DRM/Intel	display/Other	BXT CFL CNL GLK HSW ICL KBL SKL SNB	Maarten Lankhorst <bugs@mblankhorst.nl>	high	1. Marta Löfstedt <marta.lofstedt@intel.com> (41) 2. Maarten Lankhorst <bugs@mblankhorst.nl> (9) 3. Hector Velazquez <hector.franciscox.velazquez.suriano@intel.com> (3) 4. Jani Saarinen <jani.saarinen@intel.com> (3) 5. Martin Peres <martin.peres@free.fr> (2) 6. Francesco Balestrieri <francesco.balestrieri@intel.com> (1)	Oct. 9, 2017, 12:47 p.m.	never	6 days, 22:17:39 ago	7	1:42:21
fdo#103494	NEEDINFO	Inescapable system freeze on initial X startup drm/i915	DRM/Intel	GPU hang	BSW/CHT	*Intel GFX Bugs mailing list <intel-gfx-bugs@lists.freedesktop.org>*	medium	1. *aun.sswick@gmail.com* (10) 2. Jani Saarinen <jani.saarinen@intel.com> (4) 3. *omega@online.de* (3) 4. Lakshmi <lakshminarayana.vudum@intel.com> (3) 5. Mika Kuoppala <mika.kuoppala@intel.com> (2) 6. Chris Wilson <chris@chris-wilson.co.uk> (1) 7. Ville Syrjala <ville.syrjala@linux.intel.com> (1) 8. Francesco Balestrieri <francesco.balestrieri@intel.com> (1)	Oct. 28, 2017, 3:30 a.m.	87 days, 21:43:32 ago	59 days, 7:08:10 ago	60	16:51:50

Intel GFX CI

intel Software

What are the available test systems for Linux?

Name	Description	Available hardware	Results latency
0-day	Mostly build testing, Intel proprietary	Intel servers	Days to weeks
Kernel-CI	Post-merge distributed build and boot testing. Reports mostly through emails.	Any HW you might want to plug to	Minutes to hours
Snowpatch	Open source tools for running tests using Jenkins in response to emails (using patchwork).	N/A	N/A
Intel GFX CI	Build and boots, then run IGT (including a lot of suspend testing) and piglit. Picks up patches from the mailing list, sends automatic emails with the curated results. Mostly open source: fdo-patchwork, cibuglog, i915-infra	130 machines (all Intel gens starting from 2004)	30 minutes for BAT 6 hours for full results

Objectives of Intel-GFX-CI

- Provide an accurate view of the state of the HW/SW (all supported combinations).

- Results should be:
  - transparent: Should contain the full HW and SW configuration;
  - fast: Basic results in under 30 minutes, complete ones in half a day;
  - visible: make the results public and hard to miss (reply in ML);
  - stable: noise level should be zero (be aggressive at blacklisting unstable tests);

Intel GFX CI - https://intel-gfx-ci.01.org

Current state: provide timely, public, stable and transparent results for:

- Trees:
  - pre-merge: DRM-tip, IGT
  - post-merge: DRM-tip, Linus' tree, Linux-next, *-fixes, Dave Airlie's branch
- Machines (total of 130 systems / 22 different platforms (Gen 3 to upcoming Gens)):
  - GDG (Gen3, 2004) -> ICL (not released yet)
  - sharded machines: 6 SNB, 7 HSW, 10 SKL, 7 KBL, 8 APL, 9 GLK, 4 ICL
  - GVT-d BDW and SKL (Virtualization)
- Displays interfaces: HDMI, DVI, DP, eDP, DP-MST, DSI, TB, LVDS
- Test suites:
  - IGT:
    - BAT: fast-feedback: ~290 tests, ran on all machines
    - Full: KMS + some GEM tests: ~2700 tests, ran on sharded machines
  - Piglit: Run on 5 different systems during the Full test cycle
- Throughput
  - from 22k tests/day (Aug 2016) to ~3M tests/day (now)
  - bug filing: usually under half a day during working hours (700+ in 2018)

Intel-GFX CI: Let's collaborate!

- Infrastructure:
  - New community started at XDC:
    - Aims at creating an open source CI toolbox, with well defined interfaces
    - Targets having distributing testing with multiple HW-specific farms like kernel-ci
    - URL: https://gitlab.freedesktop.org/gfx-ci/documentation
  - i915 infra: https://gitlab.freedesktop.org/gfx-ci/i915-infra

- IGT:
  - Write new / improve the driver-agnostic tests
  - Write driver-specific tests for your device

- Hardware:
  - Create/modify testing-oriented hardware
  - Example: Google's chamelium which allows testing hot-plugging

Conclusion

Conclusion

CI makes upstream development easier, faster, and less buggy!

Questions / discussion

Contacts

Tomi Sarvela

- Infrastructure and most of the automation software

Arkadiusz Hiler

- IGT and FDO’s Patchwork maintainer, back up for Tomi

Martin Peres

- Ezbench and CI bug log maintainer, Bug filing

Lakshmi Vudum

- Bug filer, main bug scrubber

Petri Latvala

- IGT maintainer, Ezbench