

# KernelCI: A New Hope for Regressions

**Ana Guerrero López**

**Collabora**

October 24, 2018

## Who I am



- ▶ Free software user and enthusiast since 2001
- ▶ Debian Developer since 2006
- ▶ working at Collabora since earlier this year in KernelCI

# Presentation Outline

## **Project Overview**

What's new in the last two years

KernelCI in action: The media subsystem pilot

Future Plans

## What is KernelCI?

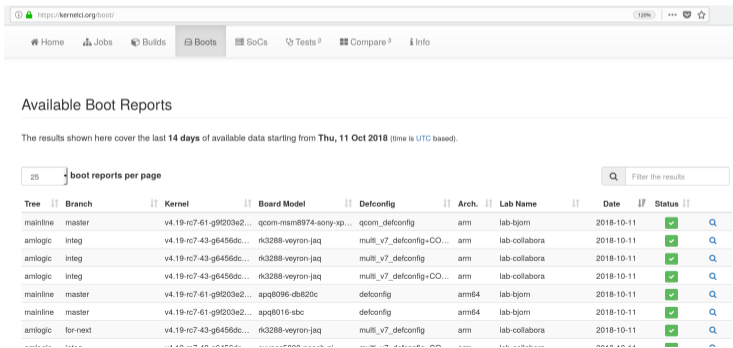
- ▶ An Open Source test automation system for the upstream Linux kernel
- ▶ All the work is done by a distributed community of developers and hardware labs
- ▶ There is a service [kernelci.org](https://kernelci.org) and a code project KernelCI

## How does KernelCI works?

- ▶ Jenkins monitors a list of kernel trees and builds them when there are updates
- ▶ Once all the kernel binaries have been built for each architecture, KernelCI asks the labs to start performing all the tests
- ▶ All the data related to the results of the boot and tests plans run is returned by labs and stored in the KernelCI backend

# How does KernelCI works? - getting results

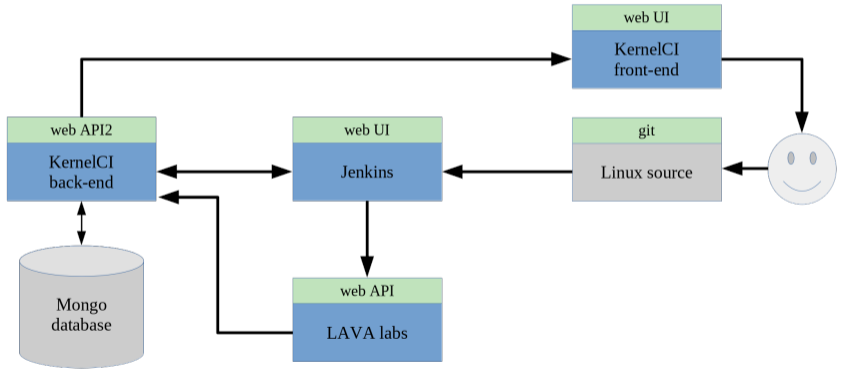
- ▶ The build logs, boot and test results are mailed to a list of pre-defined email addresses
- ▶ The ongoing results can be browsed in the KernelCI frontend <https://kernelci.org>



The screenshot shows the KernelCI website interface. At the top, there is a navigation bar with links for Home, Jobs, Builds, Boots (selected), SoCs, Tests, Compare, and Info. Below the navigation bar, the page title is "Available Boot Reports". A note indicates that the results shown cover the last 14 days of available data starting from Thu, 11 Oct 2018. There is a search bar and a filter for "boot reports per page" set to 25. The main content is a table of boot reports with columns for Tree, Branch, Kernel, Board Model, Defconfig, Arch, Lab Name, Date, and Status. The status column shows green checkmarks for successful boots and a blue magnifying glass icon for each row.

Tree	Branch	Kernel	Board Model	Defconfig	Arch.	Lab Name	Date	Status
mainline	master	v4.19-rc7-61-g9f203e2...	qcom-msm8974-sony-xp...	qcom_defconfig	arm	lab-bjorn	2018-10-11	✓
amlogic	integ	v4.19-rc7-43-g6456dc...	rk3288-veyron-jaq	multi_v7_defconfig+CO...	arm	lab-collabora	2018-10-11	✓
amlogic	integ	v4.19-rc7-43-g6456dc...	rk3288-veyron-jaq	multi_v7_defconfig	arm	lab-collabora	2018-10-11	✓
amlogic	integ	v4.19-rc7-43-g6456dc...	rk3288-veyron-jaq	multi_v7_defconfig+CO...	arm	lab-collabora	2018-10-11	✓
mainline	master	v4.19-rc7-61-g9f203e2...	apq8096-db820c	defconfig	arm64	lab-bjorn	2018-10-11	✓
mainline	master	v4.19-rc7-61-g9f203e2...	apq8016-sbc	defconfig	arm64	lab-bjorn	2018-10-11	✓
amlogic	for-next	v4.19-rc7-43-g6456dc...	rk3288-veyron-jaq	multi_v7_defconfig	arm	lab-collabora	2018-10-11	✓

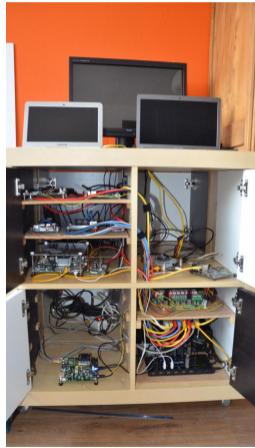
# A diagram on how KernelCI works



# Labs pictures



Cambridge



Sjoerd Simons'



Kevin Hilman's

Open First



## How does KernelCI development work?

- ▶ All the code is hosted in GitHub <https://github.com/kernelci>
- ▶ Community interactions happens mostly via four channels:
  - ▶ IRC channel in Freenode `#kernelci`
  - ▶ Mailing list <https://groups.io/g/kernelci>
  - ▶ Pull requests / GitHub tracker
  - ▶ Weekly hangout
- ▶ Testing infrastructure (staging) before merging things in production.

## Why does KernelCI matter?

- ▶ Speed up kernel development and avoid regressions in published releases
- ▶ Make LTS kernels more stable
- ▶ Advantages of community CI vs internal CI
- ▶ CI has the potential to improve a community and avoid friction

# Presentation Outline

Project Overview

**What's new in the last two years**

KernelCI in action: The media subsystem pilot

Future Plans

Open First

## Automatic bisection

- ▶ Some branches get merged with many commits
- ▶ When a boot fails, only a range of commits is known
- ▶ Goal: automatically isolate the breaking commit
- ▶ Challenges:
  - ▶ Avoid false positives
  - ▶ Trigger relevant bisections (avoid duplicates...)
  - ▶ Deploy into production
  - ▶ Provide useful results
- ▶ Learn more in this [blog post!](#)

## Automatic bisection - first results!

- ▶ Found QEMU bug on stable 4.14

<https://lists.linaro.org/pipermail/kernel-build-reports/2018-January/thread.html#27424>

- ▶ Found DRM deadlock on Samsung “Peach-Pi”

<https://lists.linaro.org/pipermail/kernel-build-reports/2017-December/thread.html#26688>

- ▶ Found 2 DRM issues on NVIDIA Tegra124

<https://lists.linaro.org/pipermail/kernel-build-reports/2017-November/thread.html#25525>

<https://lkml.org/lkml/2017/12/20/278>

## Running tests plans

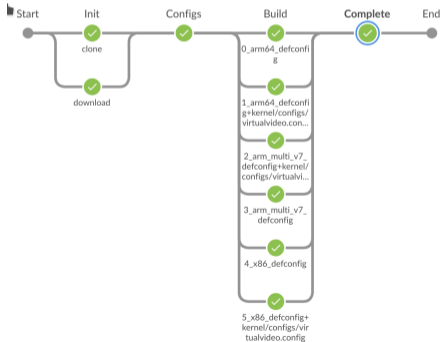
- ▶ Originally the only test available were boot tests
- ▶ Start adding test plans to further testing:
  - ▶ kselftests
  - ▶ usb, rtc, suspend/resume
  - ▶ v4l2, igt

## Reporting of test plans results

- ▶ After adding test plans, we also want to share the results
- ▶ Update the frontend to show the tests
- ▶ Send mails with the results

## Using Jenkins pipelines for jobs

- ▶ Originally all the Jenkins jobs were bash or python scripts
- ▶ Work on-going to migrate to use Jenkins pipelines and create a kernelCI library





## Automatic rootfs generation

- ▶ Standardize how rootfs used by the tests are build and generate them easily
- ▶ Using Debian (with debos) and buildroot for archs not available in Debian (arm64be, armeb)
- ▶ Easy to add new pipelines building new rootfs with new tests (igt, v4l2, etc)

## Backend updates

- ▶ Rework how data results are stored to allow multi-level tests
- ▶ Improve the import of the results from the LAVA callback
- ▶ Work not visible to users but needed to improve the quality of the information provided by kernelCI

## Documentation

- ▶ <https://wiki.kernelci.org> points to an old Linaro wiki now deprecated
- ▶ On-going documentation efforts are now in a GitHub wiki  
<https://github.com/kernelci/kernelci-doc/wiki>

# Presentation Outline

Project Overview

What's new in the last two years

**KernelCI in action: The media subsystem pilot**

Future Plans

Open First

## The idea

- ▶ Run functional tests, focused on a kernel subsystem to show what kernelCI can do
- ▶ Chosen the media subsystem because:
  - ▶ it's a fairly complex subsystem, with a large API
  - ▶ with an existing comprehensive tool to track regressions: v4l2-compliance
  - ▶ and with significant work done on virtual drivers that are able to emulate a large set of the media subsystem API

## The development plan

The plan needs to implement 3 steps in the current infrastructure:

- ▶ Setup jenkins pipeline to build rootfs images with v4l2-utils every time there is an update
- ▶ Add a v4l2 test plan running v4l2-compliance
- ▶ Setup jenkins pipelines to build the media tree every time there is an update:
  - ▶ Build kernel binaries for each architecture
  - ▶ Push the boot and test plans jobs to the labs
  - ▶ Save all the results reported by the labs in the kernelCI backend

# The result

KernelCI will send 3 mails reports: boot, build and test suites results.

```

From: "kernelci.org bot" <bot@kernelci.org>
To: ana.guerrero@collabora.com
Date: Wed, 10 Oct 2018 22:14:00 -0700 (PDT)
Subject: Test results for media/master - v4.19-rc1-275-g557c97b51336
Message-ID: <5bbdec18.1c69fb81.2d44c.3e4f@gx.google.com>

Test results for:
Tree: media
Branch: master
Kernel: v4.19-rc1-275-g557c97b51336
URL: https://git.linuxtv.org/media_tree.git
Commit: 557c97b5133669297be561e6091da9ab6e488e65
Test plans: v4l2

Summary
-----
3 test groups results

1 | v4l2 | qemu | arm | 45 total: 43 PASS 2 FAIL 0 SKIP
2 | v4l2 | rk3399-gru-kevin | arm64 | 45 total: 13 PASS 32 FAIL 0 SKIP
3 | v4l2 | qemu | arm64 | 45 total: 43 PASS 2 FAIL 0 SKIP

Tests
-----
1 | v4l2 | qemu | arm | 45 total: 43 PASS 2 FAIL 0 SKIP

Config: multi_v7_defconfig+virtualvideo
Lab Name: lab-collabora-dev
Date: 2018-10-11 03:58:27.554000
TXT log: http://staging-storage.kernelci.org/media/master/v4.19-rc1-275-g557c97b51336/arm/multi_v7_defconfig+virtualvideo/lab-collabora-dev/v4l2-qemu.txt
HTML log: http://staging-storage.kernelci.org/media/master/v4.19-rc1-275-g557c97b51336/arm/multi_v7_defconfig+virtualvideo/lab-collabora-dev/v4l2-qemu.html
Rootfs: http://staging-storage.kernelci.org/images/rootfs/debian/stretchv4l2/20180927.0/armhf/full.rootfs.cpio.gz
Test Git: git://linuxtv.org/v4l-utils.git
Test Commit: 3874aa8eb1ff0c2e103d024ba5af915b1b26f098

Test cases:
* DMABUF: PASS
* USERPTR: PASS
* MMIO: PASS
* read/write: PASS
  
```

# Presentation Outline

Project Overview

What's new in the last two years

KernelCI in action: The media subsystem pilot

## **Future Plans**

Open First



Happening now!



KernelCI is in the process of becoming a Linux Foundation project, if you're interested in joining us, please ask in the Collabora booth.

After the conference, you can mail to Gustavo Padovan and Kevin Hilman.

## In the near future

Some of the plans for the next months are:

- ▶ Improve kernelCI dashboard to help with different filtering needs
- ▶ Start tracking regressions of the tests
- ▶ Add automatic bisection when a regression in a test is detected
- ▶ Add more functional tests
- ▶ Continue improving to make the entry barrier smaller
- ▶ ...and many things more!

## In a very distant future?

What about this?

- ▶ A developer submits a patch
- ▶ CI tests the patch
- ▶ A second developer reviews and gives their +1
- ▶ A third developer reviews and gives a second +1
- ▶ The CI system merges patch
- ▶ ...and so on!

## Find more information

- ▶ Dashboard <https://kernelci.org>
- ▶ Backend API <https://api.kernelci.org>
- ▶ Organization at GitHub <https://github.com/kernelci>
- ▶ KernelCI Wiki <https://github.com/kernelci/kernelci-doc/wiki/KernelCI>

Thank you!