

KernelCI applied to distributions

LPC 2019 - *Distributions micro-conference*

9th September 2019

Guillaume Tucker

gtucker@collabora.com



COLLABORA

Open First

kernelci.org - the project

- Started in 2015 around the ARM ecosystem
- Now covering many architectures and platforms
- Taking off slowly but surely (Linux Foundation)
- Building, booting, starting to extend test coverage
- Dedicated to testing upstream kernel branches

KernelCI - the tool

- <https://github.com/kernelci/>
- Main obvious use-case is kernelci.org
- Can be used to build any kernel and run any tests
- Can be run as a dedicated instance anywhere
- What does it take to test a distribution kernel?





COLLABORA

Distro kernels vs Upstream

A few data points: Fedora

```
$ git diff --shortstat v5.3-rc7..fedora/f31
109 files changed, 118504 insertions(+), 255 deletions(-)
```

```
$ diff -u x86_64_defconfig kernel-x86_64.config | diffstat
kernel-x86_64.config | 7491 ++++++-----
1 file changed, 7221 insertions(+), 270 deletions(-)
```



A few data points: Debian

```
$ git diff --shortstat v5.3-rc7..debian  
373 files changed, 2949 insertions(+), 1200 deletions(-)
```

```
$ diff -u x86_64_defconfig config-5.2.0-2-cloud-amd64 | diffstat  
config-5.2.0-2-cloud-amd64 | 4052 ++++++  
1 file changed, 3838 insertions(+), 214 deletions(-)
```



In a nutshell

- Distro kernels carry a fair amount of patches
 - While *tracking* upstream, may not be *sent* upstream
 - Just on the edge of the kernelci.org project
- Distro kernels have large defconfigs specially tailored
 - Arbitrary config choices, may bias code coverage
 - Higher building cost per build, but fewer variants





**Can KernelCI be used to
test distribution kernels?**

Current KernelCI capabilities

- Build an arbitrary git branch with defconfigs
 - Works if all changes are as git commits
 - Works if defconfig in git tree or with small fragments
- Test plain kernel image
- Store results associated with kernel revision only



First improvements needed

- Insert steps to apply patches and generate defconfigs
 - Hooks with custom commands before build
- Packaging of resulting kernel image
 - Hooks after build: .deb, .rpm... bundle with installer
- Keep track of full OS image version in database



Extra potential improvements

- Enable testing based on user-space changes
 - Same kernel but with newer OS image
- Product-oriented features
 - Run installer, more use-space driven
- Challenge: remain compatible with kernelci.org



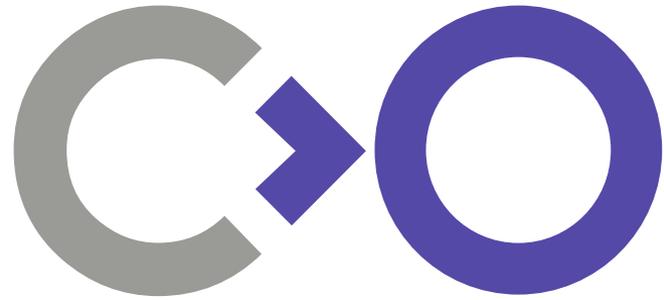


Should KernelCI be used to test distro kernels?

Some good reasons

- It seems like a pretty close match already
- Slightly different use-cases are healthy for the project
- Can help KernelCI grow to the next level
- Helps distro kernels stay close to upstream
- Distro vendors may want to contribute to the project ;)





Thank you!



COLLABORA

Open First