Introduction

- kernelci.org overview - Mark
- How does it help? - Mark
- What is LAVA - Tyler
- KernelCI LAVA V1 usage - Tyler
- Benefits of LAVA V2 for KernelCI - Matt
- Get Involved
- Questions
What is kernelci.org

- Automatic build & boot testing for the kernel
  - ARM, ARM64, MIPS and x86
  - Many kernel trees
    - Set of trees currently limited to core upstream ones (Linus’, -next and stable)
  - 260+ builds/tested commit (all in-tree defconfigs plus extras)
  - ~500 boots/tested commit over ~100 boards
  - 9000 jobs, 2,500,000 boots
  - [http://kernelci.org/stats/](http://kernelci.org/stats/)
- Results reported via mailing lists and web site
- Build hardware
  - Dedicated servers in data centres donated by Linaro, Collabora
  - Dedicated servers hosted by community members
- Boot testing labs
  - Linaro, BayLibre, Collabora, EmbeddedBits, Pengutronix, Free Electrons, Kevin Hilman, Jan-Simon Moeller, Mark, Tyler, Matt
How does it help?

● Much more likely that kernels will build…
  ○ v3.10: 53 failed configs
  ○ v3.14: 51 failed configs
  ○ v4.1: 1 failed config
  ○ Mainline: usually 0 failed configs outside of merge window
    ■ Currently higher due to MIPS
  ○ -next: this is where the errors get caught

● ...and run

● More solid basis for development

● Made merge window and early -rcs much less stressful
What is LAVA

- Job runner & scheduler for board farms
- Used by almost all kernelci.org labs
- [http://validation.linaro.org](http://validation.linaro.org) is the Linaro LAVA instance
  - [https://validation.linaro.org/static/docs/v2/#what-is-lava](https://validation.linaro.org/static/docs/v2/#what-is-lava)
- LAVA handles the deployment and booting of the kernel being tested
  - KernelCI does not need to know how to boot every board
- Developed & distributed for Debian
  - See LAVA site for other distros/source install
  - Or install in a VM or container
- For kernelci the LAVA master/UI needs to be web accessible
  - Public IP/IPv6 (static preferred, dynamic DNS can work)
  - Or host the master in the cloud
LAVA V1 Usage

- Web crawler to find kernels to boot
  - This can take several minutes with all the ARM defconfigs
- Create job files from simple templates
- Use LAVA XML-RPC api to submit jobs
- Then use XML-RPC to poll all jobs for status
  - Until all complete or timeout
  - Polling large list of jobs is heavy on LAVA resources
- Jobs that run after timeout are not cancelled and results ignored
  - Waste of Lab time
  - Labs are not always dedicated to kernelci, so may run less jobs before timeout
LAVA V2 usage - Creating LAVA jobs

● Use kernelci API to list built kernels
  ○ https://api.kernelci.org/build?job=next&kernel=next-20170301&status=PASS&arch=arm
  ○ One query gives us all the artifacts we need

● Use jinja2 templates to create job files
  ○ Supports extending/inheritance and conditionals
  ○ {% if modules_url %} - not every build produces modules

● Submit jobs to LAVA XML-RPC
  ○ But no need to poll for results
  ○ Results will come back asynchronously

● Whole create and submit process takes seconds
  ○ Does not block a jenkins build executor while polling
LAVA V2 usage - Getting boot results

- LAVA V2 uses ZeroMQ (ZMQ) for worker communication
  - We can use these messages to see when jobs are finished

- Lava-publisher daemon
  - Included in standard LAVA install
  - Labs are configured to push ZMQ messages to kernelci ZMQ listener
  - Can be limited to just kernelci jobs

- ReactOBus - [https://github.com/ivoire/ReactOBus](https://github.com/ivoire/ReactOBus)
  - ZMQ daemon that ‘reacts’ to ZMQ messages
  - React = execute script with ZMQ message data as arguments
  - Result parsing script is run when ZMQ message shows job has finished

- All jobs will add results to the API, none are wasted
  - Even if boot summary email has been sent, the API will continue getting results
LAVA V2 usage - What’s left to do

- Find a permanent place for ReactOBus daemon
- LAVA V2 device support
  - Need V2 to catch up with V1 supported devices
  - Get more labs to configure V2 devices
- Compare V1/V2 results for the same kernel on same device
  - V2 results are currently only going to the development API
Get Involved

- [https://github.com/kernelci](https://github.com/kernelci)
- Kernel engineers / Review results
  - Identify failures, root cause, bisect, submit patches upstream
- Contribute hardware
  - Build servers
  - Donate boards to existing labs
- Contribute a LAVA lab
  - Install, maintain, upstream support for your boards
- Contribute to kernelci.org infrastructure
  - Engineers to add features, work with the maintainers and extend the system
  - Or simply provide a few hosted servers (i.e. [hetzner.de](https://hetzner.de), PX61-NVME only € 71/mo)

http://kernelci.org/
irc.freenode.org #kernelci
info@kernelci.org
Questions?
Thank You

#BUD17

For further information: www.linaro.org
BUD17 keynotes and videos on: connect.linaro.org