Connected Vehicle Software Development

Automated Testing, a Collaborative Approach for the Industry

Stephen Lawrence (Renesas), GENIVI BIT Lead | May 2020

Agenda

- Shared upstream testing with a focus on **development**
  - How it complements existing in-house testing and can accelerate development
- How Genivi is contributing
  - Update on the Genivi Automated Testing Board Farm
  - The road ahead
- Q&A and discussion

Shared testing upstream

GENIVI®

Shared testing upstream

- Shared testing of important software
  - Successful example is Kernel CI
    - Community based open source distributed test automation system focused on upstream Linux kernel development.
    - Goals
      - Build every configuration for each architecture.
      - Boot these configurations.
      - Execute tests on these configurations.
- Automated Testing Summit to increase collaboration and reuse
  - Test cases – separate test cases from more abstract tooling such as scheduling
  - Interoperability – results (pass/fail, logs), components
  - Mechanisms – such as feature specific tooling, e.g. board control.

Automotive needs

- Companies have advanced internal testing setups
  - proprietary s/w which can’t be shared and
  - OSS s/w which is <– here shared testing is possible
- OSS included in company-internal testing anyway so why share?
  - Pooling resources creates ability to test wider variation of versions and configuration than is normally done in a production or internal platform project (remember Kernel CI)
  - In a complex stack trying to cover everything internally is very difficult
  - Development of test tools and test cases is time consuming and costly.
  - We use component Foo v4. Should we take v5?
    - v5 may not be tested internally (yet), but in wider community it might be
  - Investigating upstream components for integration
- Conclusion: ability to look upstream for test results is a stronger basis for development

GENIVI Contribution

Recap of what’s already in place

- Distributed CI of “systems” using GENIVI GoCD instance
  - Builds GDP and Baseline
  - Central server, with remote build agents
- GENIVI CI Policy encourages use of GitHub-integrated tools such as Travis-CI where teams select their own tooling for components
- GENIVI source hosted in GitHub
  - Integrated with GoCD to sanity build test pull requests for GDP and baseline
- Yocto Baseline (meta-ivi)
  - Meta-ivi-test image contains component unit tests
- Components
  - Mix of testing in individual companies and in the open, e.g. DLT
- New automated test initiative
  - Discussed at last AMM and now a reality
  - GENIVI LAVA Board Farm
  - Android and Linux testing

GENIVI s/w scope has expanded

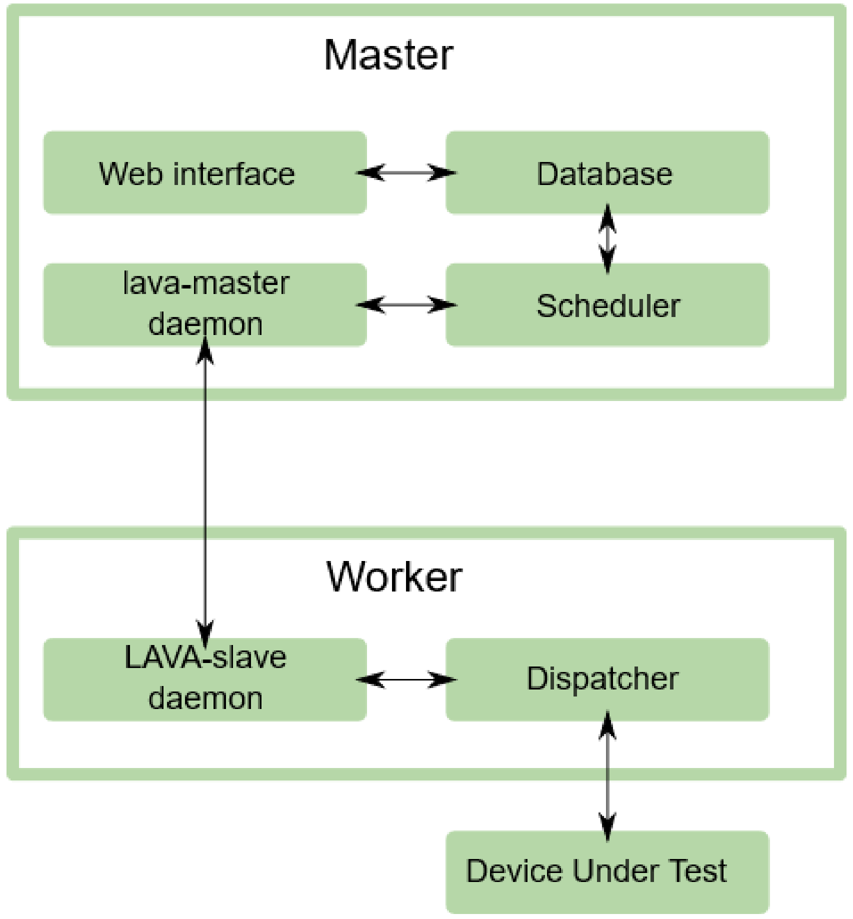
- Multi-OS strategy
  - Domain Interaction evolving into Multi-OS
  - Android SiG, Cloud and Connected Services Project etc.
  - Meaning multiple dev environments
  - Test infrastructure needs to be flexible
- Collective opportunity
  - Possibility to collaborate and integrate with existing testing infrastructures for supported development platforms such as Apertis, WebOS, Adaptive Autosar and Android.
  - Favour working together towards greater combined solutions, over repetition
  - Open test instances enhance shared development of tooling, test cases and learning
  - Open dialog and flexibility

Automated test initiative: launch recap

- Announced at last AMM
- Working mode: make a start, be flexible and open to collaboration with other orgs
- LAVA based test system to be connected to Genivi GoCD CI (and other CI as needed)
  - Distributed system in wide use.
  - Strong support for complex deployment use cases on embedded hardware.
  - Proven to scale to large deployments.
- Genivi instance will start one control server and one lab containing QEMU and automotive hardware
- Will use it for CIAT test of future GENIVI code emerging from Multi-OS

Automated test initiative: LAVA

- What is LAVA?
  - LAVA is the Linaro Automation and Validation Architecture
  - System for deploying OSs onto physical and virtual h/w to run tests.
  - Designed to automate validation during development
  - Wide device support
  - Extensive feature list
  - See Overview in LAVA documentation for full details
- Architecture
  - A LAVA instance consists of two primary components
    - LAVA Master (control server)
    - LAVA Worker (execute tests on boards) for QEMU and automotive hardware
  - YAML based test job descriptions



Automated test initiative: Genivi instance status

- Has been live and stable for some time now.
- Genivi LAVA Master (server)
- Genivi LAVA Worker (slave)
  - Renesas are hosting a lab currently containing the following DUTs:
    - QEMU
    - R-Car M3 Starter Kit
    - R-Car H3 Starter Kit with Kingfisher expansion board fitted
  - More Workers/labs wanted..
- Configuration
  - Running in Docker containers created using lava-docker from Kernel CI project
  - Leveraged work occurring in embedded industrial Civil Infrastructure Platform (CiP) (LAVA instance)
  - Worker is currently running recent LAVA release v2020.02.
    - Enables new support for handing Android host tools in Docker containers
    - Plan to update to v2020.04

Automated test initiative: Genivi instance status

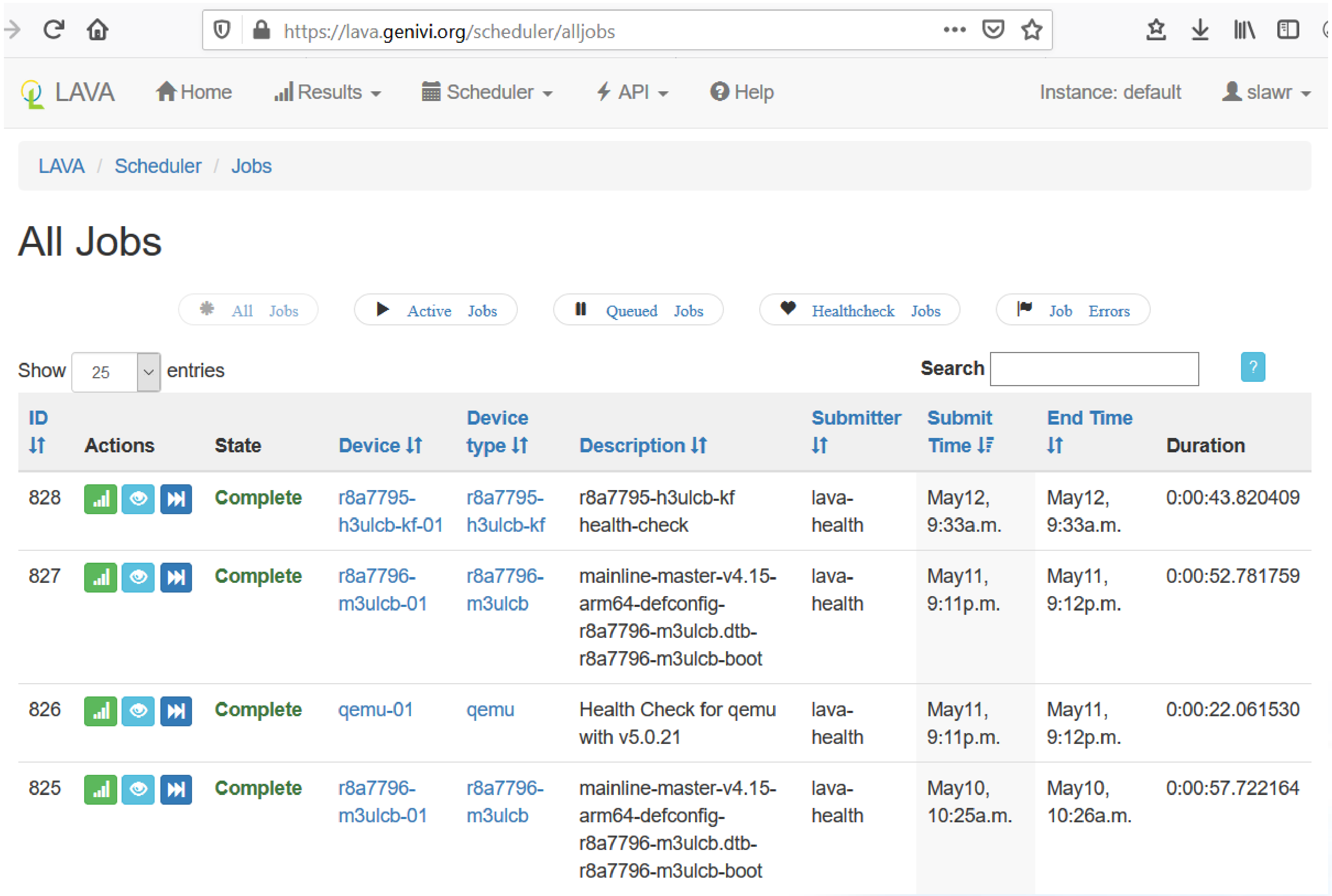
- Flexible approach to inputs
  - Allows input from different CI
  - Test artifacts can come from Genivi CI or be downloaded
- Linux
  - Linux based boot tests have been running stably for some months
  - Have proven running of meta-ivi-test unit test suite using LAVA
  - GoCD pipeline in place to execute tests. Now completing integration so meta-ivi pull requests are sanity tested against meta-ivi-test unit tests.
- Android
  - Android builds have been containerised
  - Using new features introduced in LAVA v2020.02 to handle Android host tools in Docker container
  - Have successful flashing of Android binaries using LAVA
  - Now moving to test execution.

How can we all contribute? Q&A and discussion

- The LAVA instance has been stable for some time. Now its time to make use of it.
- Contribute test cases or help with integration
- Contribute LAVA Workers
- Android test expertise
- Integrate your next GENIVI collaboration into the GENIVI CIAT
  - Easier done from the start, than later of course
- Propose other integration opportunities with existing testing infrastructure
  - Internal - what can be shared upstream?
  - Related alliances

- The board farm exists. Let's put it work and expand it.
- Q&A and discussion

LAVA Master Web-UI



All Jobs

ID	Actions	State	Device	Device type	Description	Submitter	Submit Time	End Time	Duration
828		Complete	r8a7795-h3ulcb-kf-01	r8a7795-h3ulcb-kf	r8a7795-h3ulcb-kf health-check	lava-health	May12, 9:33a.m.	May12, 9:33a.m.	0:00:43.820409
827		Complete	r8a7796-m3ulcb-01	r8a7796-m3ulcb	mainline-master-v4.15-arm64-defconfig-r8a7796-m3ulcb.dtb-r8a7796-m3ulcb-boot	lava-health	May11, 9:11p.m.	May11, 9:12p.m.	0:00:52.781759
826		Complete	qemu-01	qemu	Health Check for qemu with v5.0.21	lava-health	May11, 9:11p.m.	May11, 9:12p.m.	0:00:22.061530
825		Complete	r8a7796-m3ulcb-01	r8a7796-m3ulcb	mainline-master-v4.15-arm64-defconfig-r8a7796-m3ulcb.dtb-r8a7796-m3ulcb-boot	lava-health	May10, 10:25a.m.	May10, 10:26a.m.	0:00:57.722164

Thank you!

Visit GENIVI:

http://www.genivi.org

http://projects.genivi.org

Contact us:

help@genivi.org

GENIVI®