Fuego
LTS/LTSI Testing, Projects, and Initiatives

June 2018

Tim Bird

Fuego Maintainer

Sony Electronics

Outline

- Fuego and LTS/LTSI testing
- Fuego projects
- Industry initiatives
- Recommendations

Fuego and LTS/LTSI testing

- What does Fuego do?
- What's missing?
- How to expand testing effort?

How does Fuego support LTS/LTSI testing?

- Have lots of existing tests
  - Especially LTP and kselftest
- Have testplan_lts
  - But it needs refinement
- Makes it easy to run a set of test suites, and see results

What's missing

- Make it easy for end users to specifically test LTS kernels
  - Need to include triggers, build, provisioning, notifications for LTS
- Tests for regression-checks for specific LTS fixed bugs
- More sharing of pass-criteria (test results analysis customization)
  - Required to avoid false positives

How to expand testing effort?

- Vision: Anyone can do LTS/LTSI testing
  - 10,000 single-board nodes is better than 100 labs with 10 boards each
- Barriers:
  - Automated provisioning (kernel install)
  - Test setup is not easy enough
  - How to provide results?
  - Who will follow up on failure reports?
- Vision2: more test coverage
- TL;DR – more testers, more tests

Provisioning and scaling the testing effort

- Automated provisioning
  - Requires hardware control for 100% reliability
  - Less than 1% of users will use hardware to automate their kernel installs
  - Want to support semi-automated provisioning
- Trying hard in Fuego to avoid requiring hardware board control
- “Semi-automated” means:
  - Try software board control, and fall back to user intervention

Fuego projects

- Provisioning work in progress
- Fuego features
- More tests

Provisioning WIP

- Jenkins CI job for LTS testing
  - Helper scripts and hardware for LTS kernel testing on Ubuntu
    - LTS download and build
    - Ubuntu kernel replacement
    - boot automation with ‘ttc’
    - usb keyboard automation
      - teensy-usb – host-controlled keyboard for target
- Need an upstream for this
  - Some work checked in to ‘ttc’ board control system

Fuego Features

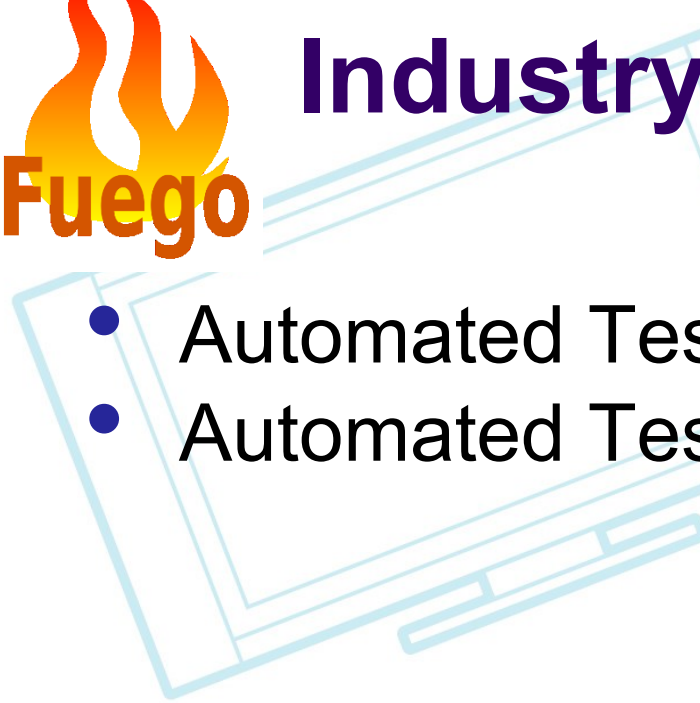
- Pre-Built docker image
  - Eliminate long Fuego install step
- Test program binary cache
  - Remove need for SDK in order to test
- Focus on pass-criteria customization and sharing
  - For testplan_lts tests, to remove false positives

More tests

- Don’t have a concrete plan here
- Fuego leverages LTP and kselftest
- Recommendation is for test authors to put new tests in those frameworks
- No organization is chartered to specifically write LTS/LTSI regression tests

Industry initiatives

- Automated Test Stack standards
- Automated Testing Summit



Test System problems

- No “lego blocks” for test system infrastructure
- Current systems are monolithic
  - e.g. Hard for Fuego to use LAVA as board control software
  - Have mismatches in models, artifacts
- Lots of islands of work
- Nobody handles off-DUT hardware orchestration
  - Maybe LAVA, but it’s not generalized
    - (e.g. LAVA multi-node tests)

Automated Test Standards

- Would be good to define:
  - objects, methods, interfaces, protocols
- Want to mix and match test stack layers, and allow separate implementations to compete
  - board control
  - test orchestration
  - results parsing
  - results aggregation
  - analysis, etc.
- Reuse features from other domains
  - e.g. log results visualization
  - e.g. libvirt for hardware board control

Test Stack standards work

- Discussions started at ELCE 2017
- http://elinux.org/Board_Farm
  - Some research on different DUT control software
- No entity chartered to define or describe layers
  - Samsung has SLAV stack definition, based on MuxPi project
- Maybe start with board control standards
  - Provisioning standards would be nice

Automated Testing Summit

- October 25, Edinburgh Scotland
  - See http://elinux.org/Automated_Testing_Summit
- Sponsored by Linux Foundation Core Embedded Linux Project
- Attempt to assemble wide variety of Linux test stakeholders and practitioners

Recommendations/Goals

- Add new tests for specific LTS commits to kselftest or LTP
- Finish conversion of kselftest to TAP
- Some group should work on Test Standards
- Continued work on Fuego usability/scalability projects

Resources

- Fuego web server:
  - http://fuegotest.org/
  - wiki: http://fuegotest.org/wiki
- Mailing list:
  - https://lists.linuxfoundation.org/mailman/listinfo/fuego
- Repositories:
  - https://bitbucket.org/tbird20d/fuego
  - https://bitbucket.org/tbird20d/fuego-core

Fuego Jamboree #2

- Saturday, June 23, 9:00 to 12:00
- Ariake, Tokyo, Japan
- Hosted by Panasonic
- Details at:
  - http://fuegotest.org/wiki/Fuego_Jamboree_2
    - Please add your name to attendee list, if you plan to come

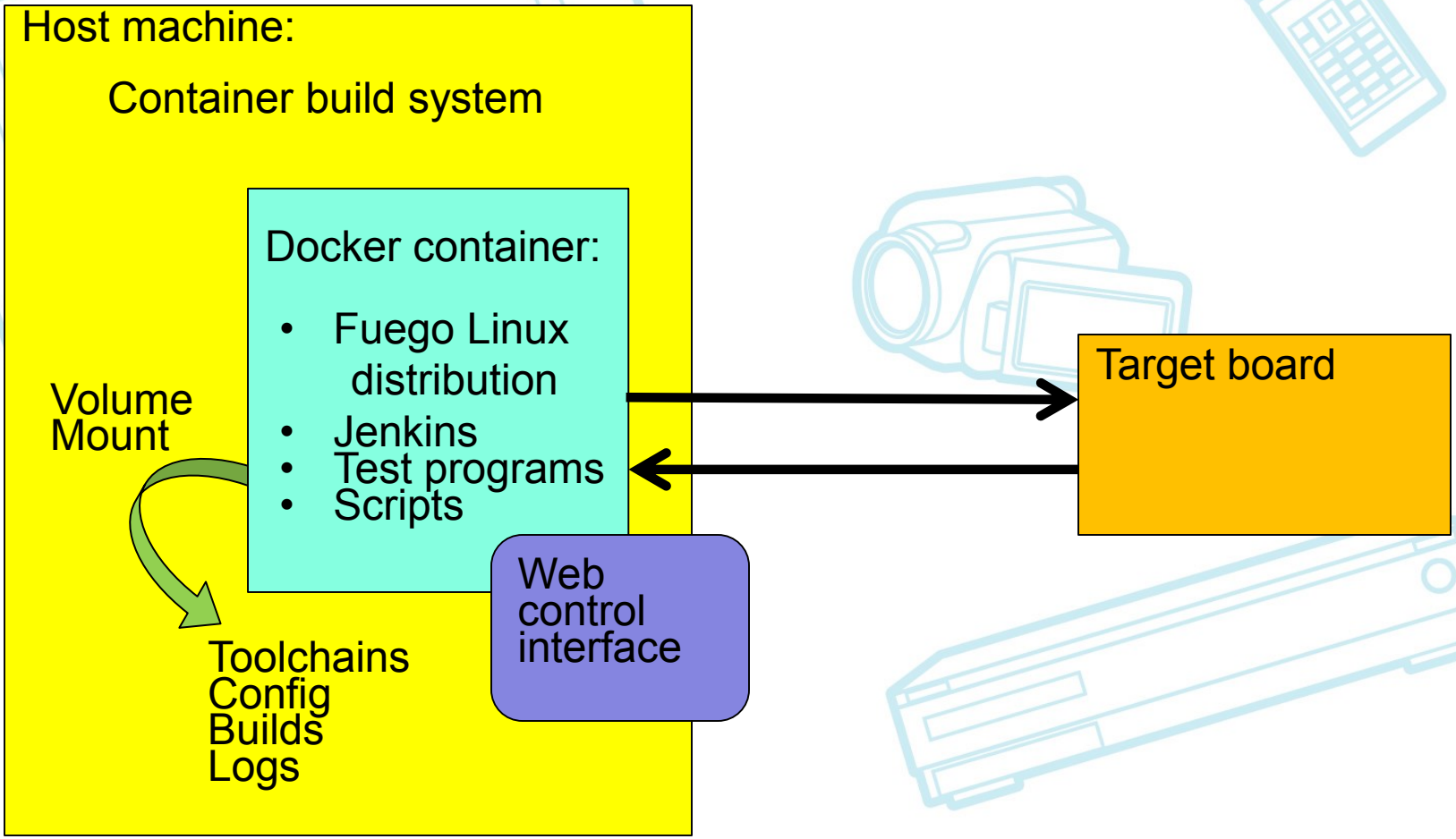

Fuego

Reference material

Micro-Introduction

Fuego =
(Fuego Linux distribution +
host scripts +
pre-packaged tests +
Jenkins)
all inside a container

Architecture Diagram



Core features

- Distribution of Linux for testing
- Build system
  - Architecture-neutral & inherently cross-platform
- Includes a collection of tests
  - Scripts for test execution
  - Results parsing, analysis, and visualization
- Report generation
- Multiple transports
- Jenkins front end
  - Also has a command line tool

Version 1.2 Features

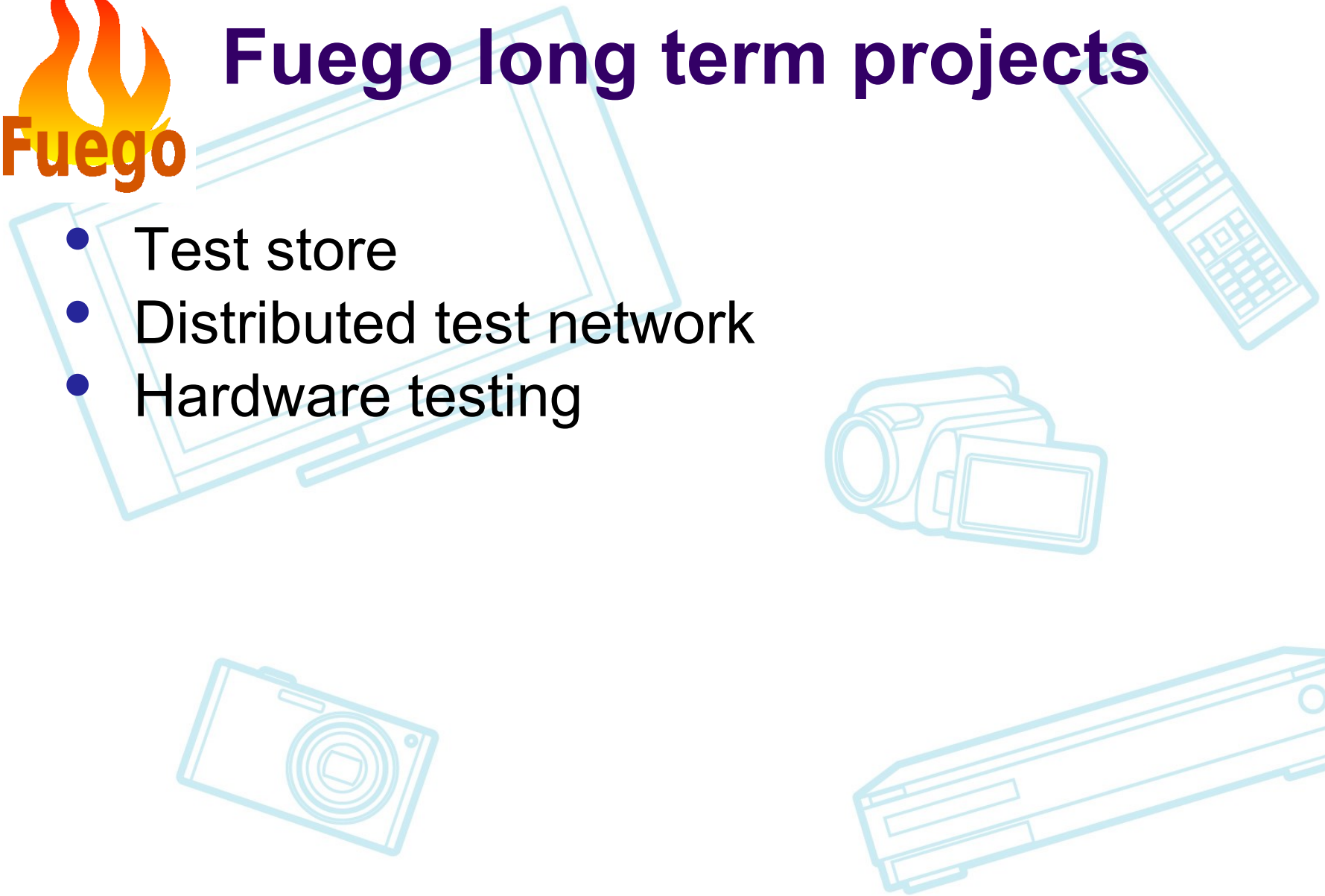
- Unified Output Format
- Test dependency system
- Complex pass criteria handling
- Dynamic board variables
- Charting
- Get test program source from git repositories
- Test improvements

Version 1.3 Features

- Report generation improvements
- Log splitting by testcase
- New tests
  - Including Fuego self-tests
- Web page and image comparison tools
- Infrastructure enhancements
  - Hardware board control
  - Individual test phases
  - ftc outside the docker container

Fuego long term projects

- Test store
- Distributed test network
- Hardware testing



Board automation standards

- Presentation at Linaro Connect
  - See http://fuegotest.org/ffiles/Test-Standards-LC-2017.pdf
- Lots of meetings at ELCE on this
  - Pengutronix introduced labgrid
  - Linutronix demonstrated r4d and libvirt
  - BOF resulted in some collaboration:
    - See https://elinux.org/Board_Farm
    - Mailing list for discussion:
      - https://lists.yoctoproject.org/listinfo/automated-testing
- Please join this discussion