

Establish an automated testing lab for AGL

Jun 21, 2018

Automotive Linux Summit, Tokyo

Liu Wenlong

Nanjing Fujitsu Nanda Software Technology.Co.,Ltd(FNST)

- Liu Wenlong(liuwl.fnst@cn.fujitsu.com)
- Linux Software Engineer (2015 ~)
- AGL CIAT Member (2017 ~)
 - focus on CIAT for AGL
 - Drivers test, LTP, Fuego and LAVA

- **Preface**
- **LAVA Quickstart**
- **Fuego Quickstart**
- **Fuego&LAVA solution**
- **LAVA tests in Fuego**
- **Future Work**

CIAT(Continuous integration automated testing)

Existing tools,

- Autotest

<https://autotest.github.io/>

- BuildStream

<https://wiki.gnome.org/Projects/BuildStream/>

- **LAVA**

<http://validation.linaro.org>

- **Fuego**

<http://fuegotest.org/>

- KernelCI

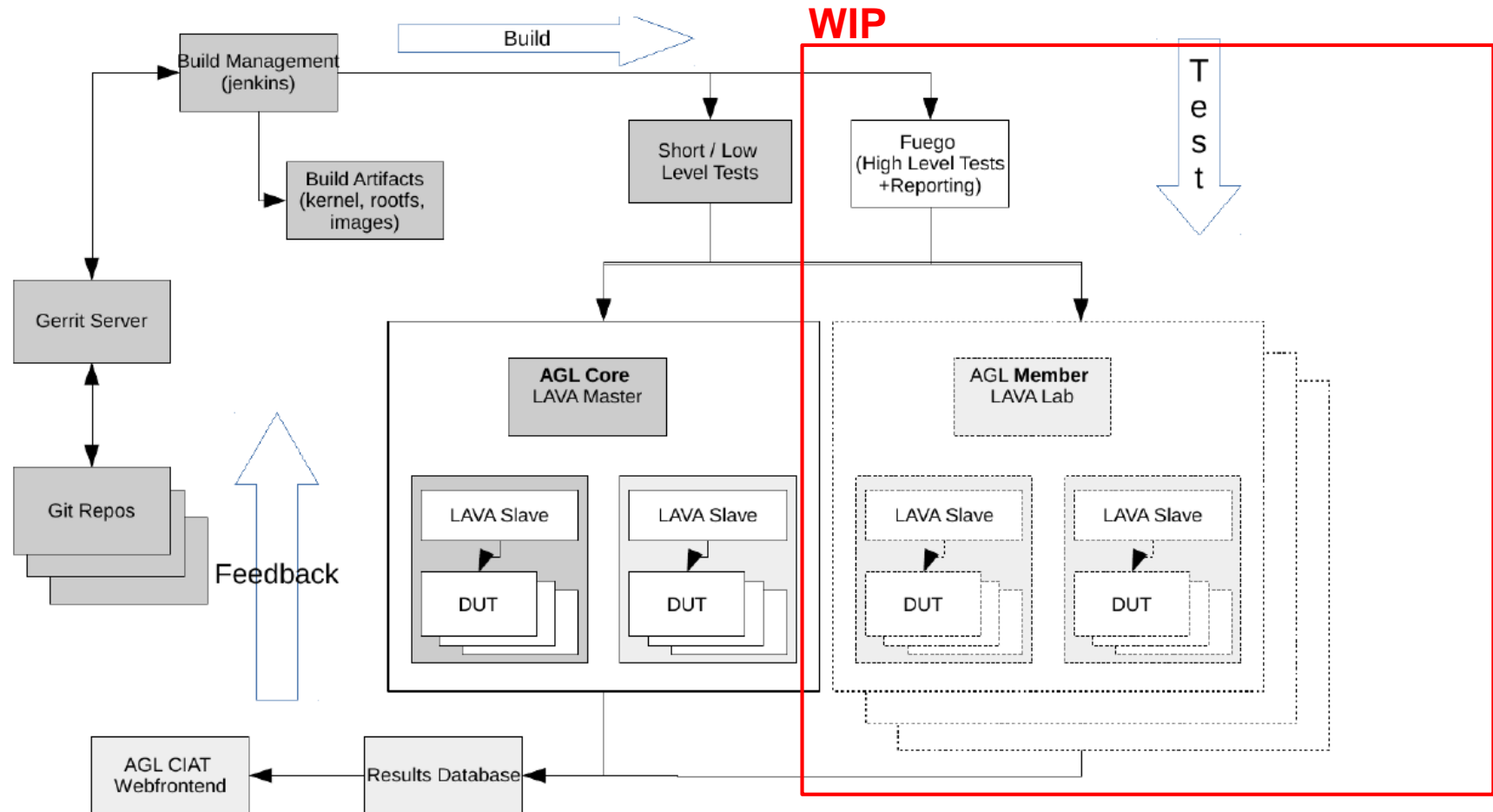
<https://kernelci.org/>

Test tools comparison,

| | LAVA | Fuego |
|------------------------------|--------|--------|
| Build farm | No | Can be |
| Pre-packaged tests | No | Yes |
| Jobs management | No | Yes |
| Reporting(Results parser) | Yes | Strong |
| Distributed lab support | Yes | No |
| Board management | Strong | Yes |
| System deployment(auto boot) | Yes | No |
| Web interface | Yes | Yes |
| Easy to extend | No | Yes |
| Easy to install | No | Yes |

Fuego+LAVA will be a good solution for our testing lab and AGL CIAT framework.

AGL testing lab(CIAT loop)



Let's see how to establish an automated testing lab.

Refer to: https://schd.ws/hosted_files/aglammeu17/5c/jsmoeller_How_to_Write_Tests_for_the_AGL_HW_Test_Infra.pdf

LAVA installation and configurations,

➤ LAVA installation,

https://www.validation.linaro.org/static/docs/v2/installing_on_debian.html

Or install with docker <https://github.com/kernelci/lava-docker> (2018.4)

➤ LAVA configurations,

- Add new devices

<https://www.validation.linaro.org/static/docs/v2/first-devices.html>

- Devices control (ser2net/PDU)

<https://www.validation.linaro.org/static/docs/v2/first-installation.html>

➤ Do LAVA tests,

- Write lava tests

<https://www.validation.linaro.org/static/docs/v2/developing-tests.html>

- Submit lava tests with “lava-tool”

<https://www.validation.linaro.org/static/docs/v2/lava-tool.html>

Let's have a look at the LAVA test definition to lay the groundwork for the later parts.

LAVA test definition (YAML file),

metadata:

```
git.branch: agl-branch
image.type: AGL
.....
```

Meta section

device_type: r8a7796-m3ulcb

actions:

```
- deploy:
  to: nbd
  os: oe
  kernel:
    url: http://xxx/Image
.....
```

Action section

- boot:

```
timeout:
  minutes: 10
method: u-boot
.....
```

Boot section

- test:

```
definitions:
- repository: https://git_qa_repo[1]
  from: git
  path: test-suites/yocto-ptest.yaml
  name: yocto-ptest
```

- test:

```
definitions:
- repository:
  .....
  run:
    steps:
      - lava-test-set start set-pass
      .....
      - lava-test-set stop set-fail
  from: inline
  name: health-test
  path: inline/health-test.yaml
```

Test section

We can easily to do different tests on different HWs by customizing different sections.

[1] <https://git.automotivelinux.org/src/qa-testdefinitions>

For details, refer to: [<https://www.validation.linaro.org/static/docs/v2/dispatcher-actions.html>]

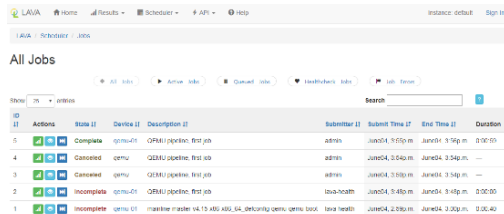
About those “sections” above, refer to:

[https://schd.ws/hosted_files/aglammeu17/5c/jsmoeller_How_to_Write_Tests_for_the_AGL_HW_Test_Infra.pdf]

LAVA connections,

Tester/developer side:

LAVA Web interface



- Write/Submit lava tests
- Check test results
- Boards management
- ...

LAVA HOST:

LAVA master
LAVA worker



Power Management

Services:
dhcpd/tftpd/nb
d/nfs/...

Serial management

Board farm:

Board Power Supply

serial/network

DUT 01



DUT 02



.....

Relay /PDU



WAN ...

LAN ...

We can use LAVA in our auto-testing lab now.

LAVA test results,

LAVA Home Dashboard **Results** Scheduler API Help Instance: DensoTEN-FNST-instance liuwl.fnst

LAVA / Scheduler / All Jobs

All Jobs

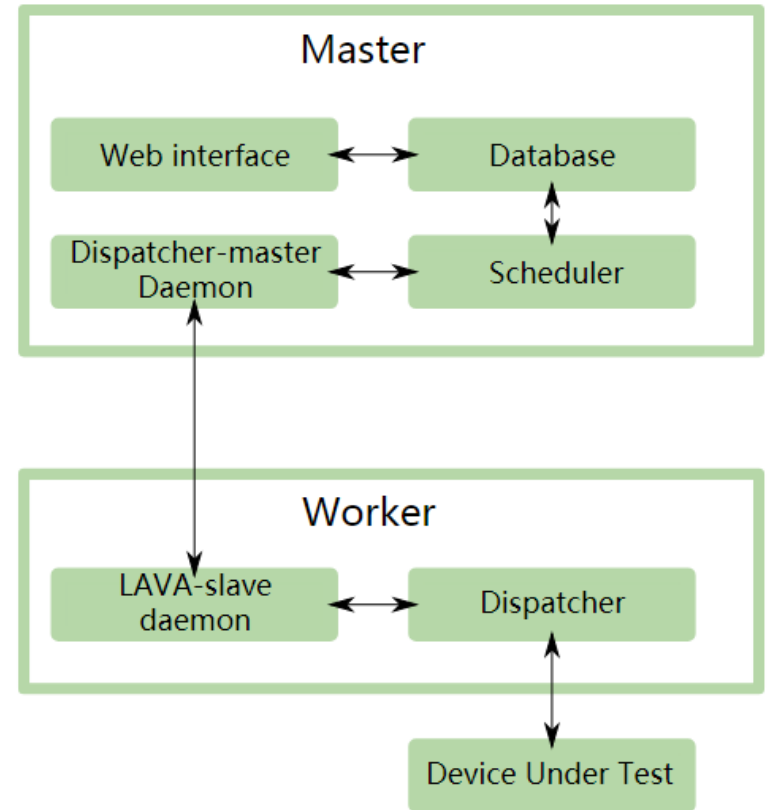
Test results in LAVA

Show 25 entries

| ID | Actions | Status | Job ID | Job Name | Priority | Host | Job Type | Start Time | End Time | Duration |
|-----|---------|----------|--------|---------------------|----------|------------|---------------------|--------------------------|-------------------------|----------|
| 336 | | Complete | 336 | lava-health | 8 | liuwl.fnst | lava-health | May 22, 2018, 10:21 a.m. | | 0:01:05 |
| 335 | | Complete | 335 | lava-health | 4 | liuwl.fnst | lava-health | May 21, 2018, 6:05 p.m. | | 0:00:55 |
| 334 | | Complete | 334 | lava-health | 8 | liuwl.fnst | lava-health | May 21, 2018, 10:20 a.m. | | 0:01:10 |
| 333 | | Complete | 333 | lava-health | 4 | liuwl.fnst | lava-health | May 20, 2018, 6:03 p.m. | | 0:00:55 |
| 332 | | Complete | 332 | lava-health | 8 | porter01 | lava-health | May 20, 2018, 10:19 a.m. | | 0:01:26 |
| 331 | | Complete | 331 | lava-health | 4 | qemu-01 | lava-health | May 19, 2018, 6:02 p.m. | | 0:02:02 |
| 330 | | Complete | 330 | lava-health | 8 | porter01 | lava-health | May 19, 2018, 10:17 a.m. | | |
| 329 | | Complete | 329 | lava-health | 4 | qemu-01 | lava-health | May 18, 2018, 6:01 p.m. | | |
| 328 | | Complete | 328 | liuwl.fnst | 1 | porter01 | liuwl.fnst | May 18, 2018, 4:12 p.m. | | |
| 327 | | Complete | 327 | AGL-short-smoke-wip | Medium | m3ulcb01 | AGL-short-smoke-wip | May 18, 2018, 4:06 p.m. | May 18, 2018, 4:08 p.m. | 0:02:02 |

LAVA features and diagram,

- Testing changes on multi-HW
- Boot testing, bootloader testing and system level testing
- Not a test lab
- Not a build farm
- Not a complete CI solution
- Not a set of tests
- Distributed test lab
- Support various bootloaders



Source from <https://www.validation.linaro.org/static/docs/v2/first-installation.html>

For details, please refer to,

[\[https://www.validation.linaro.org/static/docs/v2/contents.html\]](https://www.validation.linaro.org/static/docs/v2/contents.html)

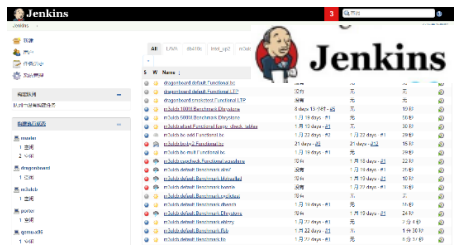
Fuego installation and configurations,

- Fuego installation steps
http://fuegotest.org/wiki/Fuego_Quickstart_Guide
- Add a board
http://fuegotest.org/wiki/Adding_a_board
- Add a toolchain
http://fuegotest.org/wiki/Adding_a_toolchain
- Add a test
http://fuegotest.org/wiki/Adding_a_test
- How to write good tests?
<http://fuegotest.org/wiki/Presentations>

Fuego connections,

Tester/developer side:


Jenkins web interface



Users can,

- Job management
- Run tests
- Check test results

Fuego Server:



Jenkins:

- job management
- job trigger
- test results display
- etc

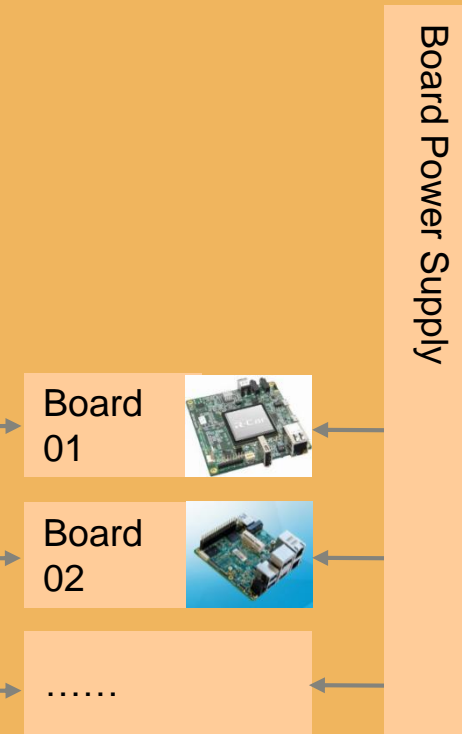
File server:

- share test logs
- documents
- etc

Fuego Framework

- test tarball management
- test scripts (build/run/analysis)
- boards/toolchain configuration
- ...

Board farm:



Board Power Supply

Board 01

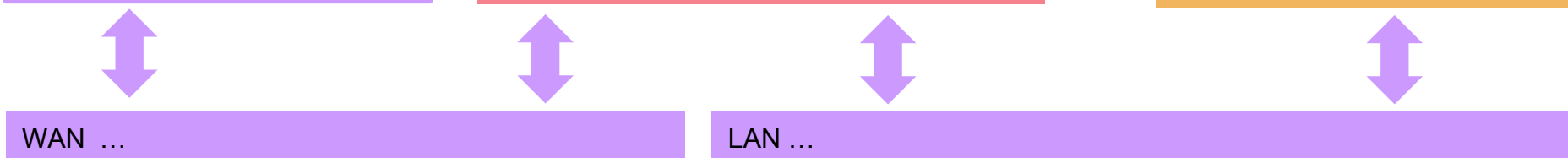
Board 02

.....

SSH/...

SSH/...

SSH/...



We can use Fuego in our auto-testing LAB now.

Fuego test results(report),

The screenshot shows the Jenkins web interface. On the left, there are navigation menus for '新建', '用户', '任务历史', '编辑视图', '删除视图', and '系统管理'. Below these are sections for '构建队列' (Build Queue) and '构建执行状态' (Build Execution Status). The main area displays a list of jobs under 'Intel-UP2.Functional'. One job, 'intel-up2.default.Functional.fuse', is highlighted with a red box. A red arrow points from this job to a detailed view on the right. This detailed view shows the test results for 'Project intel-up2.default.Functional.fuse' on the 'm3ulcb-Functional.fuse-default' board. It includes a table of results for build numbers 1, 4, 5, and 6, and a 'Totals' section. A red box highlights the 'fuse' row in the table, which shows 'FAIL' for build 1 and 'PASS' for builds 4, 5, and 6. Below this, another detailed view for 'intel-up2-Functional.fuse-default' is shown, with a red box highlighting its 'fuse' row, which shows 'PASS' for all build numbers 1 through 10. A red arrow points from the 'fuse' row in the second table to a 'Detailed testcase outputs' section on the left, which contains the shell script for the test case.

Detailed testcase outputs

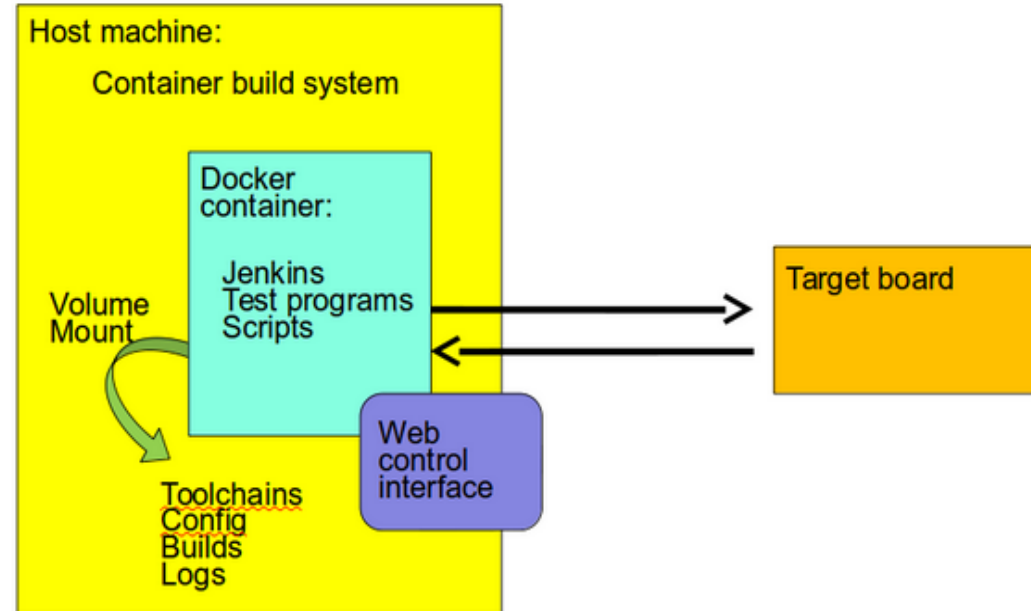
```
if ./hello $test_path/fuse/test_hello ; then
  echo 'TEST-1 OK'; else echo 'TEST-1 FAIL';
fi;
TEST-1 OK
if cat $test_path/fuse/test_hello/hello | grep "Hello World!";
  echo 'TEST-2 OK'; else echo 'TEST-2 FAIL';
fi;
Hello World!
TEST-2 OK
if ./fusemount -u $test_path/fuse/test_hello; then
  echo 'TEST-3 OK'; else echo 'TEST-3 FAIL';
fi;
TEST-3 OK

mkdir -p $test_path/fuse/test_fioc
if ./fioc -o allow_other -o sync_read -o nonempty \
  -o intr -o big_writes -o remember=1 -o kernel_cache \
  -o kernel_cache $test_path/fuse/test_fioc; then
  echo 'TEST-4 OK'; else echo 'TEST-4 FAIL';
fi;
TEST-4 OK
if ./fusemount -u $test_path/fuse/test_fioc/; then
  echo 'TEST-5 OK'; else echo 'TEST-5 FAIL';
fi;
TEST-5 OK
```

Fuego Quickstart

Fuego features and diagram,

- Highly customizable
- Unified test outputs
- Flexible test configuration
- Running tests in batches
- Pre-packaged tests
- Jenkins based
- Do tests with command lines
- Board setup is simple & flexible



Source from <http://fuegotest.org/wiki/Architecture>

For details, please refer to, <http://fuegotest.org/wiki/Architecture>

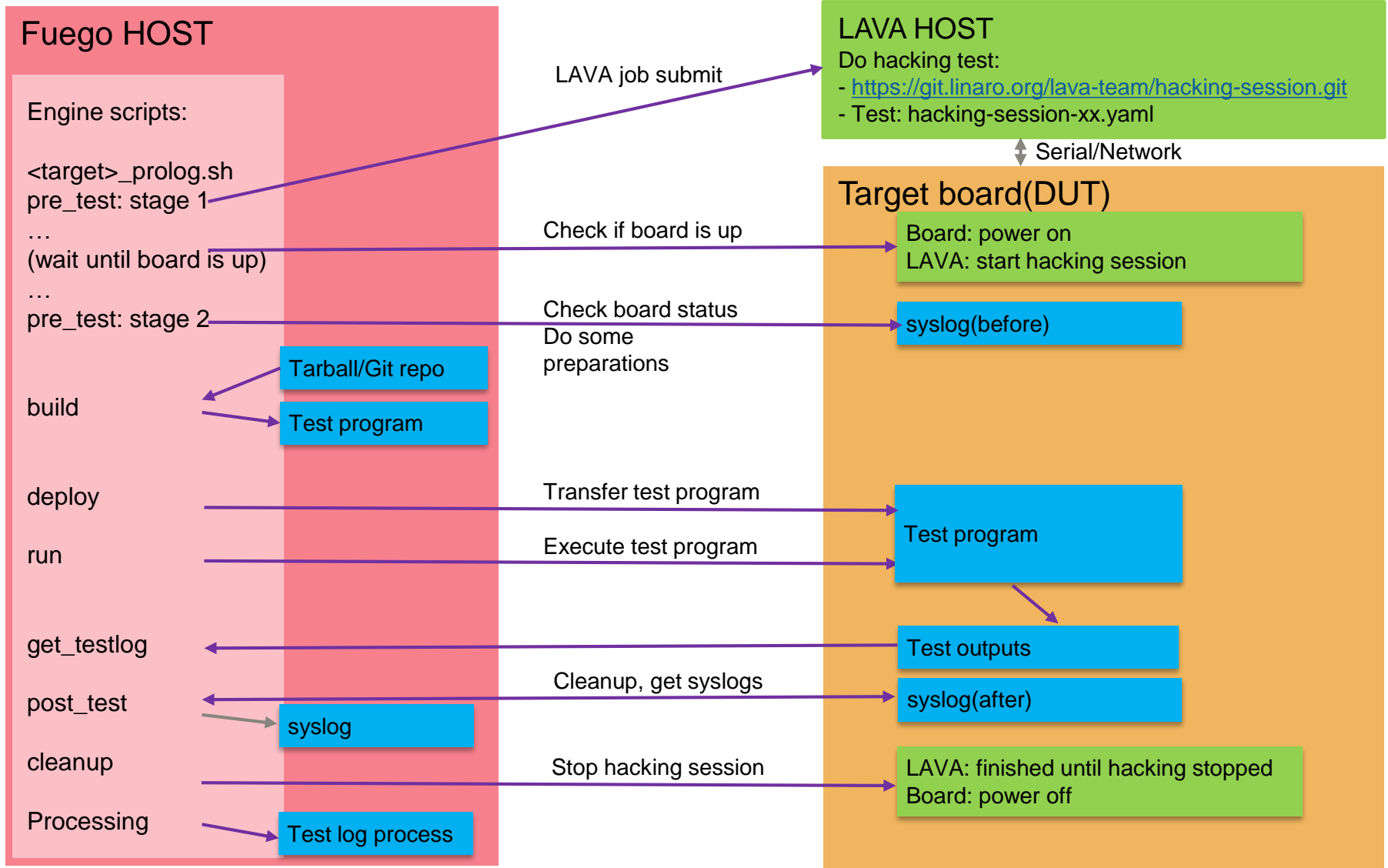
To do more with LAVA and Fuego,

- If we want to do LAVA+Fuego test, how?
 - Use LAVA as a board handler layer (existing feature)
 - Do LAVA job submit within Fuego
 - Enhanced board management for Fuego
 - LAVA board lab can be used in Fuego
 - Fuego has the ability to do OS upgrade
 - Use Fuego as LAVA test wrapper and build farm (new)
 - Do specified OS deployment/upgrades on real hardware automatically
 - With the help of Fuego, LAVA has the ability to make test results meaningful
 - Over 100+ Fuego pre-packaged tests can be used in LAVA
 - Fuego has the ability to do tests with remote board lab

Let's see the existing “Fuego+LAVA” solution first.

Fuego&LAVA solution

Existing “Fuego+LAVA” feature(work flow),



Refer to, http://fuegotest.org/wiki/Architecture#fuego_test_phases

Existing “Fuego+LAVA” feature(quickstart),
(example with “Intel UP2” board)

- Enable the following variables in board file:
 - TARGET_SETUP_LINK="fuego-lava-target-setup"
 - TARGET_TEARDOWN_LINK="fuego-lava-target-teardown"

- Add a “upsquared.lava” and a “upsquared.lava.yaml”:
 - file “upsquared.lava” holds env variables
credentials and values for KERNEL,ROOTFS, LAVA_HOST, etc
 - file “upsquared.lava.yaml” holds a template for the lava test job

We can do Fuego+LAVA tests on Intel UP2 board with those steps above.

Refer to: <https://bitbucket.org/tbird20d/fuego/commits/7d6a953455ada9b79d8a3a7ede523cd79647de5c>

Fuego&LAVA solution

Existing “Fuego+LAVA” feature(test results),
(example, “Funcinal.bc” test on Intel UP2 board)

Jenkins > upsquared.default.Functional.bc >

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Project upsquared.default.Functional.bc

upsquared-Functional.bc-default

| board: upsquared | | | | |
|----------------------------------|--------------|------|------|------|
| test set: default | | | | |
| kernel: 4.9.56-intel-pk-standard | | | | |
| test case | results | | | |
| | build_number | | | |
| | 2 | 3 | 8 | 10 |
| bc | PASS | PASS | PASS | PASS |

```

+ tee .myjob
+ lava-tool submit-job http://liuwl.fnst@192.168.0.1/RPC2 testjob.yaml
submitted as job id: 162
++ sed -e 's/ submitted as job id: ##g'
++ grep 'submitted as job'
++ cat .myjob
+ export MYJOB=162
    
```

Console outputs

Build History

[trend](#)

find

- #10 10-Jun-2018 16:30
[testlog](#) [run.json](#)
- #9 10-Jun-2018 16:24
[testlog](#) [run.json](#) [fuegoqlog](#) [devlog](#) [prolog.sh](#)
- #8 10-Jun-2018 16:12
[testlog](#) [run.json](#)

162 **Complete** Medium upsquare-01 Fuego tests with LAVA support liuwl.fnst June14, 10:33a.m. June14, 10:39a.m. 0:06:10

New “Fuego+LAVA” feature,

About the current “Fuego+LAVA” solution, it CANNOT,

- Not support remote board lab
- Cannot save test duration
- Cannot customize LAVA tests

We have those demands above, so how?

- Add a new job to generate yaml test files to support remote board lab
 - To use AGL “relog-scripts”[1] will be a nice choice
- LAVA tests customization(do multi-tests at one DUT boot)
 - add new tests in AGL qa-testdefinitions repo[2] to yaml test files

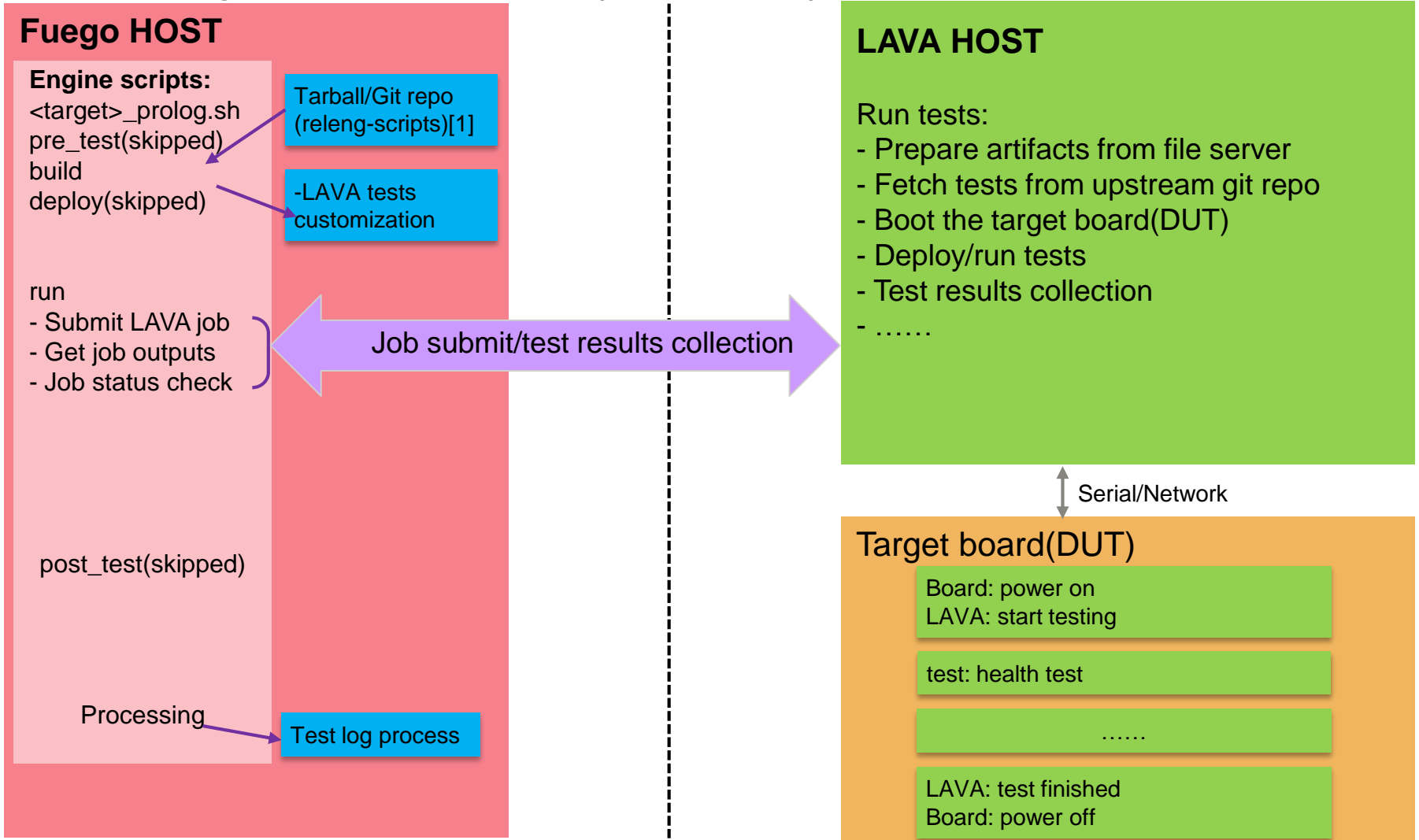
Let’s have a look at the new feature next.

[1] <https://git.automotivelinux.org/AGL/relog-scripts>

[2] <https://git.automotivelinux.org/src/qa-testdefinitions>

Fuego&LAVA solution

New “Fuego+LAVA” feature(work flow),



[1] <https://git.automotivelinux.org/AGL/relog-scripts/>
 Refer to, http://fuegotest.org/wiki/Architecture#fuego_test_phases

New feature: add a new test to generate yaml test files,

- Add a new test, (named “Functional.lava” in this example)
 - Download the releng-scripts[1] to “Functional.lava/releng-scripts.tar.gz”
- LAVA tests generation and customization,

```
$ cat /fuego-core/engine/tests/Functional.lava/fuego_test.sh
tarball=releng-scripts.tar.gz
```

```
.....

function test_build {
    # generate lava yaml test file.
    ./utils/create-jobs.py --machine $NODE_NAME --test ${FUNCTIONAL_LAVA_TESTS} ¥
    --boot ${FUNCTIONAL_LAVA_BOOT_TYPE} ¥
    --url ${FUNCTIONAL_LAVA_FETCH_URL} > test.yaml
    .....
}
```

Those test specified here will be added to yaml test file.

Now, AGL only support health-test, smoke,yocto-ptest in upstream repo[1].
But we can specify whatever other tests from github or somewhere and add it to yaml test file.

[1] <https://git.automotivelinux.org/AGL/releng-scripts/tree/templates/tests>

New feature: jobs submit and results reporting,

- Job submit with lava-tool,

```
$ lava-tool submit-job https://${LAVA_USER}@${LAVA_HOST} test.yaml
```

```
# lava-tool can also be used to check the job status and get test outputs from LAVA.
```

- LAVA outputs analysis,

```
- {"dt": "2018-05-07T10:50:41.001315", "lvl": "debug", "msg": "Received signal: <TESTSET> START set-pass"}
- {"dt": "2018-05-07T10:50:41.001630", "lvl": "info", "msg": "Starting test_set set-pass"}
- {"dt": "2018-05-07T10:50:41.062658", "lvl": "target", "msg": "<LAVA_SIGNAL_STARTTC always-pass>"}
- {"dt": "2018-05-07T10:50:41.063024", "lvl": "target", "msg": "<LAVA_SIGNAL_ENDTC always-pass>"}
- {"dt": "2018-05-07T10:50:41.063352", "lvl": "target", "msg": "<LAVA_SIGNAL_TESTCASE TEST_CASE_ID=always-pass RESULT=pass>"}
- {"dt": "2018-05-07T10:50:41.063636", "lvl": "target", "msg": "+ lava-test-set stop set-pass"}
- {"dt": "2018-05-07T10:50:41.063917", "lvl": "target", "msg": "<LAVA_SIGNAL_TESTSET STOP>"}
- {"dt": "2018-05-07T10:50:41.064194", "lvl": "target", "msg": "+ lava-test-set start set-fail"}
- {"dt": "2018-05-07T10:50:41.064466", "lvl": "target", "msg": "<LAVA_SIGNAL_TESTSET START set-fail>"}
- {"dt": "2018-05-07T10:50:41.065469", "lvl": "debug", "msg": "Received signal: <STARTTC> always-pass"}
- {"dt": "2018-05-07T10:50:41.066349", "lvl": "debug", "msg": "Received signal: <ENDTC> always-pass"}
- {"dt": "2018-05-07T10:50:41.067118", "lvl": "debug", "msg": "Received signal: <TESTCASE> TEST_CASE_ID=always-pass RESULT=pass"}
- {"dt": "2018-05-07T10:50:41.067514", "lvl": "results", "msg": {"case": "always-pass", "definition": "0_health-test", "result": "pass", "set": "set-pass"}}
```

LAVA outputs

Easy to analyse and split the test outputs from LAVA.

Test name and result can be easily generated by the following key-value in the test outputs,

- "case": testcase
- "result": result
- "set": testset

Test results with new “Fuego+LAVA” feature(Fuego side),

Jenkins | m3ulcb.default.Functional.lava

返回面板 | 状态 | 修改记录 | 工作空间 | 立即构建 | 删除 Project | 配置

Project m3ulcb.default.Functional.lava

工作区 | 最新修改记录

Build History

| Build # | Time |
|---------|------------------|
| #30 | 2018-6-13 下午4:39 |
| #29 | 2018-6-13 下午3:20 |

Received signal: <TESTSET> STOP
Closing test_set set-pass
Received signal: <TESTSET> START set-fail
Starting test_set set-fail
+ lava-test-case always-fail --shell false
<LAVA_SIGNAL_STARTTC always-fail>
<LAVA_SIGNAL_ENDTC always-fail>
<LAVA_SIGNAL_TESTCASE TEST_CASE_ID=always-fail RESULT=fail>
+ lava-test-set stop set-fail
<LAVA_SIGNAL_TESTSET STOP>
+ set +x
Received signal: <STARTTC> always-fail
Received signal: <ENDTC> always-fail
Received signal: <TESTCASE> TEST_CASE_ID=always-fail RESULT=fail

Test case outputs in detail

m3ulcb-Functional.lava-lava

| test case | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|--------------------|------|------|------|------|------|------|------|------|------|------|
| 0_health-test | PASS | PASS | PASS | PASS | PASS | - | PASS | PASS | - | PASS |
| 1_bc-test | PASS | PASS | PASS | PASS | PASS | - | PASS | PASS | - | PASS |
| auto-login-action | PASS | PASS | PASS | PASS | PASS | - | PASS | PASS | - | PASS |
| bootloader-overlay | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| http-download | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| http-download-1 | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

LAVA tests can be triggered and its testcases can also be parsed and displayed in Fuego.

Test results with new “Fuego+LAVA” feature(LAVA side),



LAVA / Results / Test job 371 / Suite 0_health-test

**LAVA test results(LAVA side)
i.e. health-test**

Results for test suite 0_health-test - Test Job 371

Exports ?

Test suite export : CSV or YAML

Show 25 entries

| Name ↓↑ | Test Set ↓↑ | Result ↓↑ |
|-------------|-------------|-----------|
| always-pass | set-pass | ✓ pass |
| always-fail | set-fail | ✗ fail |

Bug links

```
case: always-fail
case_id: 1945
definition: 0_health-test
result: fail
set: set-fail

Received signal: <TESTSET> STOP
Closing test_set set-fail
<LAVA_SIGNAL_ENDRUN 0_health-test 291.1.3.2.4.1>
<LAVA_TEST_RUNNER>: 0_health-test exited with: 0
0_health-test-1515554789
<LAVA_TEST_RUNNER>: exiting lava-test-runner
Received signal: <ENDRUN> 0_health-test 291.1.3.2.4.1
Ending use of test pattern.
Ending test lava.0_health-test (291.1.3.2.4.1), duration 0.17

case: 0_health-test
case_id: 1946
definition: lava
duration: 0.17
namespace: common
path: inline/health-test.yaml
metadata: {'description': 'Inline test to validate test framewrok health', 'format': 'Lava-Test Test Definition 1.0', 'scope': ['functional'], 'yaml_line': 24, 'os': ['debian'], 'name': 'inline-test'}
run: {'steps': ['lava-test-set start set-pass', 'lava-test-case always-pass --shell true', 'lava-test-set stop set-pass', 'lava-test-set start set-fail', 'lava-test-case always-fail --shell false', 'lava-test-set stop set-fail'], 'yaml_line': 30}
yaml_line: 23
result: pass
revision: unspecified
uuid: 291.1.3.2.4.1
ok: lava_test_shell seems to have completed

always-fail:
  result: fail
  set: set-fail
always-pass:
  result: pass
  set: set-pass
```

test outputs

Fuego results in the next slide.

LAVA tests in Fuego

What is LAVA test,

The generic lava tests,

- Usually hosted in git repo[1]
 - yaml file describing the (set of) test(s) and scripts to execute the test(s)[2]
- Tests from,
 - from inline/git

How can we do Functional/Benchmark tests with LAVA? i.e. Functional.LTP.

- Scenario 1 : Pre-installed LTP on DUT (i.e. agl-demo-platform-qa)
- Scenario 2: LTP not installed on the DUT (i.e. agl-demo-platform)

Different solution will be used for different scenarios.

Let talk about this in the next slide.

[1] <https://git.automotivelinux.org/src/qa-testdefinitions>

[2] https://schd.ws/hosted_files/aglammeu17/5c/jsmoeller_How_to_Write_Tests_for_the_AGL_HW_Test_Infra.pdf

Functional/Benchmark tests for LAVA. (Example, Functional.LTP)

- About scenarios 1 & 2, yaml test files from git/inline can be as below[1],

```
.....
params:
  TESTS: math
  .....
run:
  steps:
    - if [ ! -e /opt/ltp ]; then wget FUEGO_BUILD_SERVER/TARBALL && tar --strip-components=1 -xzf TARBALL; else
      cd /opt/ltp; fi
    - .....
```



The TARBALL above is the pre-built LTP binaries(built in Fuego with ftc tool).

- About scenarios 2,
 - test from tar will be a better solution,
BUT, LAVA said “Support is planned for tar and url”[2].
 - LAVA transport support will also be a good solution(more Fuego style)
(WIP)

[1] <https://git.linaro.org/qa/test-definitions.git/tree/automated/linux/ltp>

[2] <https://www.validation.linaro.org/static/docs/v2/actions-test.html#from>

LTP test with “Fuego+LAVA” solution(Fuego side),

The screenshot shows the Jenkins interface for the LAVA test project 'porter.ltp_smoke.Functional.lava'. On the left, there is a 'Build History' section with a search bar and a list of builds from #18 to #39. The main area shows 'related connections' with links to the last build, last failed build, last unsuccessful build, and last completed build. On the right, there are two test result tables.

porter-Functional.lava-health-test

| test case | results | | | | | | | | | | | | | | |
|-------------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | build_number | | | | | | | | | | | | | | |
| | 32 | 33 | 34 | 36 | 37 | 38 | 39 | 54 | 55 | 56 | 59 | 60 | 61 | 62 | 63 |
| always-pass | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| always-fail | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL | FAIL |
| Totals | | | | | | | | | | | | | | | |
| pass | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| fail | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| skip | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| error | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

LTP Test results

porter-Functional.lava-LTP-test

| test case | results | | | | | | |
|---------------|--------------|------|------|------|------|------|------|
| | build_number | | | | | | |
| | 33 | 34 | 37 | 38 | 39 | 54 | 55 |
| abs01 | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| atof01 | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| float_bessel | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| float_exp_log | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| float_iperb | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| float_power | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| float_trigo | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| fptest01 | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| fptest02 | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

LAVA tests in Fuego

LTP test with “Fuego+LAVA” solution(LAVA side),

[LAVA](#)
[Home](#)
[Dashboard](#)
[Results](#)
[Scheduler](#)
[API](#)
[Help](#)
Instance: DensoTEN-FNST-instance

Results for test suite 1_LTP-test - Test Job 291

Exports [?](#)

Test suite export : [CSV](#) or [YAML](#)

Show entries

| Name ↑ | Test Set ↑ | Result ↑ | Measurement ↑ | Units ↑ | Logged ↓ | Bug Links |
|---------------|------------|----------|---------------|-------------------------|----------------------|-----------|
| abs01 | — | ✓ pass | — | LTP test results | 05/16/2018 2:56 p.m. | [0] |
| atof01 | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| float_bessel | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| float_exp_log | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| float_iperb | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| float_power | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| float_trigo | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| fptest01 | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| fptest02 | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |
| nextafter01 | — | ✓ pass | — | — | 05/16/2018 2:56 p.m. | [0] |

Future Work

About LAVA+Fuego:

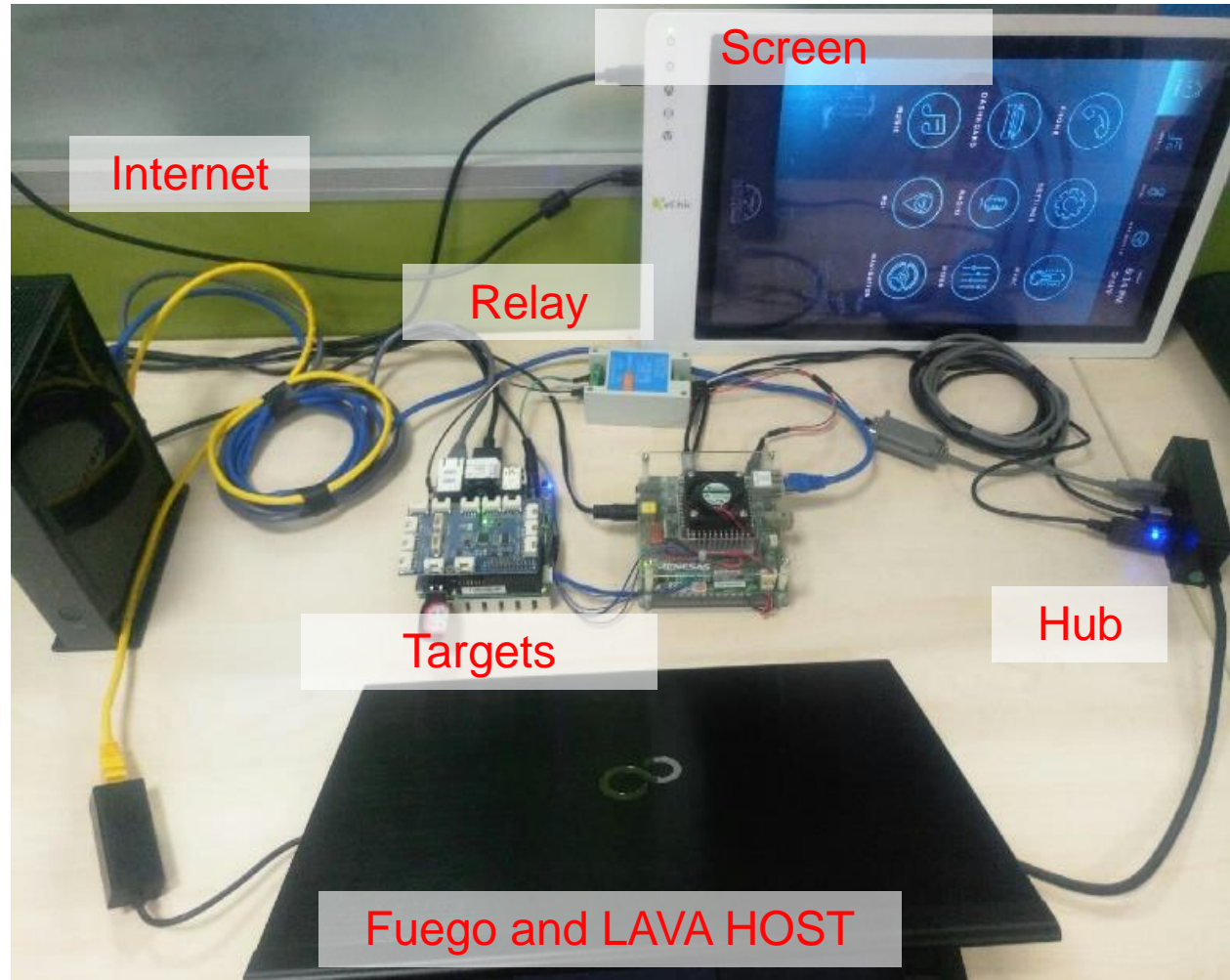
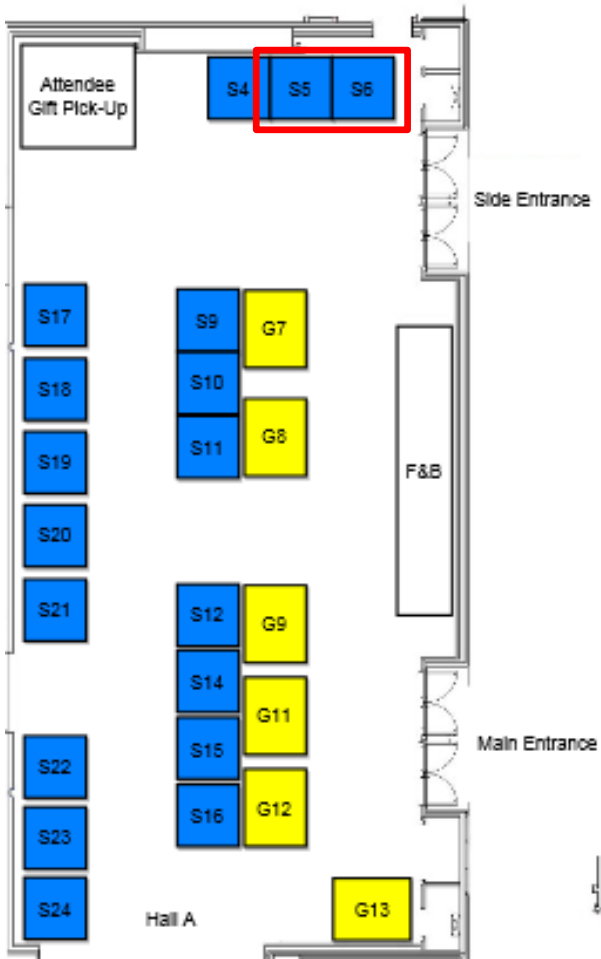
- Run build phase separately and make tar for built artifacts
- Avoid false positive for test outputs from LAVA
- Watch the new features about LAVA.
- Take part in “LAVA Transport Support” related works
(Tim Bird/Daniel Sangorrin/Jan-Simon are working on this)

OTHERS:

- New tests for AGL qa-testdefinition
- Benchmark test results displayed in html table
- Fuego failed tests investigation for R-Car M3 and Intel UP2
- Kernel LTSI test(BSP tests, drivers tests, OSS tests)


Introduction of Demo showcase

Welcome to our Demo Showcase...



Thank you!

liuwl.fnst@cn.fujitsu.com



FUJITSU

shaping tomorrow with you