Testing a large testing software

Rémi Duraffort, Linaro Ltd.
remi.duraffort@linaro.org
Who am I?

- Rémi Duraffort
- Senior Software Engineer at Linaro
- LAVA Architect
- OSS developer since 2007
  - VLC media player
  - v8 js engine
  - PRoot/CARE
  - LAVA, lavacli, meta-lava, DummySys, lavafed, ...
LAVA
A brief introduction
LAVA

- Linaro Automated Validation Architecture
- Test execution system: testing software on real hardware
  - Deploy, Boot and Test
- Usages
  - Boot testing: kernelci
  - System level testing: lkft
  - Power consumption
  - Benchmarks
  - Multinode
    - Test with many devices
  - ...

linaro.org/lava
Testing without LAVA

% power on board
% telnet localhost 2000
<enter>
=> dhcp
=> setenv serverip 10.3.1.1
=> [...]
=> bootm 0x01000000 - 0x03f00000
[...]
raspberry pi3 login: root
# run-test.sh
[...]
% power off board

Power control
Serial relay
tftp&nfs server
Testing with LAVA

Job Configuration

dispatcher

Power control
Serial relay

tftp&nfs server
LAVA architecture

Users

LAVA

server

dispatcher 1

dispatcher 2

dispatcher N

Power control

Serial relay

tftp&nfs server

dispatcher 1

dispatcher 2

dispatcher N
LAVA roles

Server

- Web UI and API
  - Submit jobs
  - Results, logs, ...
- Access control
  - Users, groups
- Scheduling jobs
  - Priority
  - Multinode jobs
- Store job logs
- Send notifications

Dispatcher

- Deploy resources
- Power on/off DUTs
- Send commands
- Parse logs
  - Kernel panic
  - Bootloader error
- Classify errors
  - Infrastructure, Bug
  - Job, Test, ...
Supported methods

deploy:
- tftp
- nbd
- flasher
- fastboot
- mps
- ssh
- docker
- vexpress ums
- ...

boot:
- cmsis
- dfu
- uboot
- pyocd
- fastboot
- docker
- qemu
- grub
- iso installer
- ...

test:
- git repository
- interactive
- minimal
- multinode
Supported device-types: 197
Why testing a testing software?
Why testing a testing softwares?

- Should be **reliable**
  - False positives
    - Lose trust in the CI
      - Developers **ignore** the CI
  - False negatives
    - Not reporting errors
    - Shipping **buggy** software
- **Bugs** and **regressions**
  - Like every software
- **Complexe** softwares
Testing LAVA
Testing LAVA

- While developing
  - Manual testing
  - ./ci-run
- Each merge request
  - Gitlab CI
- Once a day
  - meta-lava
  - federated testing
  - staging.v.l.o
Testing LAVA: GitLab CI

- Test (unittest)
  - Server: 561 tests
  - Dispatcher: 388 tests
  - On Debian 9, 10 and 11
- Analyze
  - Black
  - Static analysis: pylint, bandit
- Build
  - Debian pkg (9 and 10)
  - Docker images (amd64 and AArch64)
  - Doc
Testing LAVA: GitLab CI

● LAVA server
  ○ Just a django application
    ■ Insert data
    ■ Run some tests
● LAVA dispatcher
  ○ Board interaction
  ○ Master-slave protocol
  ○ Difficult to test
    ■ Use pytest mocking
Testing LAVA: GitLab CI

- Is it useful
  - Found many issues

- Is it enough?
  - NO!
Testing LAVA
meta-lava
Testing LAVA: a combinatorial issue

197 boards
× 16 deploy
× 26 boot
× 4 test

327,808 combinations*

* Most case are useless but that’s still huge
Testing LAVA: meta-lava

● **Goal**
  ○ Testing the full system
    ■ Including board interaction
  ○ Without any boards
  ○ Fast & cheap

● **Solutions**
  ○ Board emulation
    ■ CPU intensive
    ■ Expensive & slow
  ○ System mocking
Testing LAVA: system mocking
Testing LAVA: system mocking

- Users
- Dispatcher 1
- Server
- Power control
- Serial relay
- TFTP&NFS server
Testing LAVA: system mocking

- Users
- dispatcher 1
- Power control
- Serial relay
- tftp&nfs server
- server
Testing LAVA: system mocking

- Power control:
  - Just a command line
    - /bin/true

- Serial relay:
  - Board output
  - LAVA commands
    - DummySYS

- tftp & nfs:
  - Should check the files
    - tftp and nfs commands
Testing LAVA: system mocking

- **DummySYS**
  - Output
    - Like a real board
  - Input
    - Expect the right sequence
      - Fail if the sequence is changed
  - Use tftp & nfs resources
    - Download kernel/dtb/ramdisk
    - Mount nfs rootfs
      - Checksum some files
Testing LAVA: system mocking

Users

dispatcher 1

Power control

Serial relay

tftp&nfs server

DummySYS

server
Demo
Testing LAVA: system mocking

- Meta-lava
  - Server docker container
  - Dispatcher docker container
    - With DummySYS
- Testing master every morning
  - 27 board types
  - Including board that I’ve never seen
  - Testing board failures
    - bootloader errors
    - dhcp failing
    - ...

Testing LAVA
Benchmarks
Running 500 jobs in parallel?

- **Hardware:**
  - 500 boards
  - ~100 servers
  - Many people to plug everything
  - Some board and server will fail

- Not reliable

- **Mock some part of the system**
  - Use only one server
  - Mock the right part
    - Keep the test effective
Running 500 jobs in parallel?

- Users
- LAVA
- server
- dispatcher 1
- dispatcher 2
- dispatcher N
- Power control
- Serial relay
- tftp&nfs server
Running 500 jobs in parallel?
Running 500 jobs in parallel?

dispatcher

lava-slave

lava-run

Power control

Serial relay

tftp&nfs server

lava-master

lava-logs

... server
Running 500 jobs in parallel?

- lava-slave
- lava-run
- dispatcher
- lava-master
- lava-logs
- server
- ...

Diagram:
- Connections between components:
  - lava-slave to lava-run
  - dispatcher to lava-slave
  - lava-master to lava-logs
  - server to lava-master

...
Running 500 jobs in parallel?

Diagram showing the components:
- **lava-master**
- **lava-slave**
- **lava-run**
- **lava-logs**
- **server**

Connections:
- Dispatcher to lava-slave
- Logs from lava-logs to server
Running 500 jobs in parallel?

- Looks like lava-run
  - Same command line
  - Handle signals
  - Return value
  - Sending logs
  - Right format
  - Similar speed
- Not using CPU/RAM/IO/…
- Every services running normally
Conclusion

● System mocking is fun
  ○ Test on fake hardware
    ■ Even hardware that you don’t own
  ○ Run benchmarks
  ○ ...
● Can find many bugs that unittest won’t
● Not that difficult
  ○ Be creative!
  ○ Look at the boundaries
Question?