Building, Deploying and Testing an Industrial Linux Platform

SZ Lin (林上智)
MXcore, Software Supervisor
06/01 2017
About Me

SZ LIN (林上智)

- Software Engineer at Moxa
  - Industrial Grade Linux Distribution
- Debian Developer
- Blog - https://szlin.me
Industrial Linux Platform
Industrial/ Harsh Environments

Including smart rail, smart grid, intelligent transportation, factory automation, oil & gas, marine, and more
Industrial Linux Platform

Application

Edge Connectivity
- Serial Connectivity
- I/O Connectivity
- Video Connectivity

Industrial Computing
- Embedded Computers
- Industrial Ethernet

Network Infrastructure
- Industrial Wireless LAN
- Industrial Routers

Device
Industrial Linux Platform

Application

Software

Hardware

Device

Industrial Linux Platform

Industrial Linux Platform

Embedded Computers

Industrial Computing

Industrial Linux Platform

Industrial Linux Platform

Serial Connectivity

I/O Connectivity

Video Connectivity

Edge Connectivity

6 Hardware

Software
Industrial Linux Distribution

- Bug fixes
- Security fixes
- New userspace program backports
- New kernel features backports

Over 10+ years

Maintenance release
Ecosystems for Industrial Linux Distribution

CI: Continuous Integration
CD: Continuous Delivery/Deployment
LT: Long-term Test
Continuous Integration and Continuous Delivery/Deployment
Ecosystems for Industrial Linux Distribution

CI : Continuous Integration
CD : Continuous Delivery/ Deployment
LT : Long-term Test
CI/CD Automatic Release Pipeline

1. Building
2. Deploying
3. Testing
4. Release
CI/CD Automatic Release Pipeline

1. Building
2. Deploying
3. Testing
4. Release
Test Cases Management - Jenkins

Static analysis #1

Static analysis #2

…

Static analysis #n
THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."

HEY! GET BACK TO WORK!

COMPILING!

OH. CARRY ON.
Distributed Compiler

• Software
  – Icecream/ IceCC was created by SUSE based on distcc [5][6]
    • Improve performance of compile jobs in parallel
    • Add dynamic scheduler of the compilation jobs
    • Support multiple platform
    • Support cross compiling

• Hardware - for each node
  – SSD
  – Large capacity memory
  – Gigabit LAN
Internal/ External Developers

1. Send command

2. Get source

3. Send log & image

Webhooks

git

Local Branch

Patches

ICECC - Distributed Compiler

Client

Server

Worker #1

Worker #n
CI/ CD Automatic Release Pipeline

Continuous Integration

Code

Building

Continuous Delivery

Deploying

Testing

Release

1

2

3

4
Continuous Delivery – LAVA [7][8]

LAVA

Validation is key to the delivery of robust code and is an ongoing process involving multiple teams, covering four components:

- Device Automation
- Build Automation
- Test Planning
- Developer feedback

Device Automation is the process of providing remote access to a set of devices in order to deploy systems for testing, automate the operation of the test and collect the results. Device automation involves the provision of hardware and the software to automate tests on that hardware. The hardware provision for the Cambridge Lab is managed by the Linaro LAB team (login required) who work closely with the LAVA software team to provide the device automation for the Cambridge lab and the LAVA software for other instances inside and outside Linaro.

Build Automation is the process of preparing files which are based on developer activity and which can be deployed on devices to validate whether the developer changes have improved or broken the ability of the device to perform the required test. Linaro has the Builds and Baselines team (see the portal also – login required) to support automated builds of releases and test support files.

Test Planning involves developers and test writers identifying which tests are relevant to the development work and how at least some of those can be automated. The Test Plan needs to arrange that:

- tests are initiated automatically by developer activity,
- updated test files are prepared,
- a test job is submitted to the device,
- results are collected and made available to developers
1. Send job file via XML-RPC

2. Dispatch job via ZMQ

3. Download image via curl or wget

4. Boot up via Ethernet remote I/O

5. Deployment via TFTP

6. Send test command

Dynamic Program Analysis
- gcov [9]
- valgrind [10]
- profiling tools [11]
- ...

Platform Test
- LTP [12]
- Security testing [13]
- Kselftest [17]
- ...

Server

Master

Worker

Worker

Worker

DUT Clusters
CI/CD Automatic Release Pipeline

Continuous Integration

1. Building

Continuous Delivery

2. Deploying
3. Testing

Continuous Deployment

4. Release
CI/CD Automatic Release Pipeline

- Merge
- Pass
  - Maintainer Approval
    - Y (Pass)
    - N (Notification)

- Image Deployment
- 24/7 Long-term Platform Test

Master
Develop
Long-term Test
Ecosystems for Industrial Linux Distribution

CI : Continuous Integration
CD : Continuous Delivery/ Deployment
LT : Long-term Test
* Test cases are managed by LAVA
24/7 Long-term Platform Test

Endurance test
Compatibility test...

Longevity
Long-term support at least 10 years life cycle with bug fixes, new features and new hardware components

Robustness
Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input [18]

Reliability
Reliability is enhanced by features that help to avoid, detect and repair hardware faults [1]

Security
Quick response in resolving CVE/vulnerabilities and attacks in platform
24/7 Long-term Platform Test

**Longevity**
Long-term support at least 10 years life cycle with bug fixes, new features and new hardware components

**Robustness**
Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input [18]

**Reliability**
Reliability is enhanced by features that help to avoid, detect and repair hardware faults [1]

**Security**
Quick response in resolving CVE/vulnerabilities and attacks in platform

Fuzz testing [14][15][16]
...

Confidential
24/7 Long-term Platform Test

Power failure test
Reboot test
...

Reliability
Reliability is enhanced by features that help to avoid, detect and repair hardware faults [1]

Security
Quick response in resolving CVE/vulnerabilities and attacks in platform
24/7 Long-term Platform Test

Daily test for CVE...

Security
Quick response in resolving CVE/vulnerabilities and attacks in platform
CI/ CD/ LT are concepts of software engineering instead of tools or procedures
Future Work

- Collaboration with upstream developer
- Bootloader test integration
- Performance/ Unit test integration
- Test cases framework enhancement
  Keep eyes on Fuego
Q & A
Thank You
References

References

[14] https://github.com/google/syzkaller
[16] https://github.com/google/oss-fuzz
[17] https://kselftest.wiki.kernel.org
Meet Our Team

Elvis Yao
Software engineer at Moxa System.

Fero Zhou
Software engineer at Moxa System.

Gavin Lai
Software engineer at Moxa Networking.