OAI 5G Core Network Testing Framework

Lionel Gauthier / Raphael Defosseux

June 22nd, 2018

5G software alliance for democratising wireless innovation
Test: Fundamental Part of Continuous Integration

• Continuous Integration is just a tool

• The objectives are:
  • Ease of merge all contributors’ effort
  • **Validate** correctness and functionality of new code
  • Keep old feature/code validated
  • Improve Code Quality

• Scripts on remote machine allow
  • Verifying rules and build
  • Running a test-suite or only a sub-set
  • Provide feedback in order to help decision
Generic OAI CI Work Flow

1. Code Changes → GitHub
2. Integration Job - Triggered by Pull Requests
3. Jenkins Server
4. Release Job - Triggered by Commits on develop / master
5. Pipeline Stages
   - Guidelines Check
   - Static Code Analysis
   - Build Most Variants
   - Test with Equipment
   - Guidelines Check
   - Static Code Analysis
   - Build All Variants
   - Full Test with Equipment
6. FOCUS
   - BEAUTIFUL
   - CORRECT
   - FUNCTIONAL
   - TESTED
7. Report
     - In Release Notes

Feedback on status - ✔️ ✗
Why The Notifications in the Repository?

• It helps integrators to make a decision
  • The decision will remain human
    • Jenkins / GitHub plugin allows automated acceptance of Pull Requests

• It also provides traces / proof that a release has all quality indicators
  • That’s for the Release Job
Decentralization is Possible

- OAI Alliance members/partner can duplicate the Core Network bench
- N CI test-beds with or without overlapping test coverages
  - Could speed up the CI process
Topology

- Git Server is
  - Hosted by GitHub
- Jenkins Server is
  - Hosted by Eurecom
  - Visible from WWW
  - Configured only by admins
- Testing node is
  - Hosted by Eurecom
  - In Private Network
  - Primary OS is Ubuntu
Virtualization

• Testing Node (orion):
  • 48 cores: Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
  • 64 Gbytes RAM
  • 900 Gbytes on “/” (mount size)
  • Ubuntu 16.04.4

• As a starting point we will use a VM for each Network Function (vNF)
  • VM could use any Linux-flavored OS (Fedora, Red Hat, ...)
    • Initial CI setup is still Ubuntu-based
Template Virtual Machines

• We are pre-building Virtual Machines that will be used as `Template`:
  • Starting from distribution ISO Images
  • Starting from pre-build Cloud Images
• Template can be also dedicated to a single target vNF
• We perform a minimal installation and credentials link:
  • So the ‘Jenkins’ user can logged onto VMs without any visible password
• To optimize the disk usage:
  • We are using QCOW2 format
  • Snapshots are introduced in the test process
  • All created VM for a given build will be deleted at the end of run
CI Job Workflow: Building the vNFs

Pipeline Groovy script → Retrieve Code / Scripts → Clone on the fly → Pre-build Template VM → vNF → Build / Configure NF in each cloned VM
Jenkins View: Building the vNFs
Optimization: apt-cacher

• During the build process, we are installing packages and compiling some libraries

• If your organization does not have mirrored archives server
  • Install apt-cacher on the test node
  • It speeds up the download of packages
NG4T Tester: NG40-RAN-S1-X2-DT
Testing step: vNF lifetime

• Stop NG40 corresponding vNF
• Modify NG40 framework configuration to incorporate OAI vNF to test
• Start OAI vNF
• Run test-suite
• Stop OAI vNF
• Put back original NG40 framework configuration
• Start again NG40 corresponding vNF
Testing OAI MME vNF

NG40-RAN

OAI MME vNF

NG40 SGW vNF

NG40 PGW

NG40 HSS vNF
Testing OAI HSS vNF
Testing OAI SPGW vNF
Testing the complete OAI Core Network

- Shutdown NG40 tester
- Configure NG40 framework with all OAI vNFs
- Start all OAI vNFs, NG40 Tester
- Run test suite
- Stop all vNFs
- Put back original NG40 framework configuration
Testing all OAI vNFs
End of CI Build Job

- Stop NG40 tester
- Stop all OAI vNF
- Archives testing results
- Destroy all images
Current Status

code changes

Feedback on status

Jenkins Server
- Integration Job - Triggered by Pull Requests
- Release Job - Triggered by Commits on develop/master

Pipeline Stages
- Guidelines Check
- Static Code Analysis
- Build Most Variants
- Test with Equipment

Pipeline Stages
- Guidelines Check
- Static Code Analysis
- Build All Variants
- Full Test with Equipment

FOCUS
- BEAUTIFUL
- CORRECT
- FUNCTIONAL
- TESTED

Implemented
Coming Soon

GitHub
web-hooks

Report In Release Notes

Integrated

BEAUTIFUL
CORRECT
FUNCTIONAL
TESTED

Guidelines Check
Static Code Analysis
Build Most Variants
Test with Equipment

Guidelines Check
Static Code Analysis
Build All Variants
Full Test with Equipment

OAI 5G Core Network Test Framework

2018-06-22
Thank you for listening