OpenAirInterface
Continuous Integration / Continuous Deployment / Continuous Delivery Training

Raphael Defosseux
December 3rd, 2019

5G software alliance for democratising wireless innovation
$ whoami

• 20+ years of experience in developing HW and SW code
• 14 years in real SW dev as
  • Coder in C/C++/Java/Android/any scripting language
  • Integrator and maintainer for industrial products
  • Negotiator of features/improvements with customers

• Joined OpenAirInterface Software Alliance on April 3rd 2018
  • As “Software Manager” → Continuous Integration and Methodology expert
About this training session

• This session is an extension of the training done during the Sophia Workshop in December 2018 and New-Jersey Workshop in June 2019
  • Updated/New slides will have a new sticker
  • Improvements have been made to the different scripts and scenarios

• This session is also an extension of the training done during the Beijing Workshop in June 2018 by Kentaro Sawada from FUJITSU
  • His presentation is available on the OAI web-site:
    • http://www.openairinterface.org/?page_id=2816

• Continuous Integration is a living animal like the OAI code
  • Some points in this presentation might quickly become obsolete
Introduction to CI

CI = Continuous Integration
CD = Continuous Deployment
OpenAirInterface Software Alliance Goals

• Provide a clean, working software base
  • Functional with a feature list
  • Verified with test-cases and coverage

• Welcome new contributors and new users
  • Ease of use

• Make life easier for
  • Contributors
  • Integrators
Why do we do Continuous Integration?

• Making sure existing features are always tested
  • That their performances are not degraded
• Making sure new feature introductions do not break existing features
  • In addition new tests have to be added to keep testing these new features
• Making sure new contributions meet quality standards
• Making integrator life easier
  • Focus is now on the code review and not validating the added functionality
Why CI is good for common users?

• Our project is always lagging on documentation
  • Even if we are pushing contributors to add in-repo doc/tutorials
    • Easier to maintain

• We have a great support amongst the community
  • Through our user / developer mailing lists

• But CI can be also a support material for a common user
  • All the scripts / configuration files are visible
  • All run / test / build logs are available
  • Perfect examples on how we run OAI
Code Quality Definition

• Code shall be:
  • “Beautiful” ➔ Does code follow coding guidelines?
  • “Correct” ➔ Does code compile and won’t crash?
  • “Functional” ➔ Is code tested against known pattern?
  • “Fully Tested” ➔ Are all code lines reached?

• Code Quality Indicators shall be
  • **Measured** by automated tools such as Continuous Integration
  • Logged as proof — **Traceability**
OAI CI Work Flow

Code Changes → Jenkins Server

- Integration Job
  - Triggered by Merge 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

BEAUTIFUL  CORRECT  FUNCTIONAL  TESTED

Feedback on status

GitHub

GitLab

Report In Release Notes
Testing

• Test pattern / scripts written by the feature developer is nice
• But it does not prevent misunderstanding on the feature implementation

• 3rd party tool / equipment is better used to test part of OAI code
OAI Code vs 3rd party

- LTEBOX
- RAN Emulator
- COTS-UE
- OAI CN
- OAI eNB / gNB
- OAI UE
- OAI CN CI
- OAI RAN CI
- OAI FLEXRAN CTL
In the Future: OAI Continuous Deployment

- ORCHESTRATION
- OAI CN
- OAI eNB / gNB
- OAI FLEXRAN CTL
- COTS-UE
- OAI UE
RAN vs CN CI Status

- Fully integrated to GitLab Merge-Request process
- CI scripts embedded in repo
- Formatting Rule Check
- Static Code Analysis
- Variant Builds on several OS
- Testing w/ simulator and RF-board
- Feedback to contributor by email

- Fully integrated with GitHub done
  - For openair-cn and openair-cn-cups
- CI scripts in a private separate repo
- Build and Configuration of vNF
  - On separate Virtual Machines
- Minimal feedback to GitHub
- No test integration w/ NG4T RAN emulator yet
- Feedback to contributor by email
How to Contribute

OAI Upstream Repositories

Keep our workspace sync'ed upstream
- `git fetch origin -prune`
- `git pull origin develop`
- `git merge origin/develop`

Make your contribution public
And create a Merge Request
The official OAI CI will run
- `git push origin my_branch`

Contributor Local Workspace

Your Upstream Repositories

Push your branch to our local central repo and do local CI
- `git pull your_origin my_branch`

- `git push your_origin my_branch`

**NEW**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>git clone oai.org/openinterface5g.git</code></td>
<td>Clone upstream repository</td>
</tr>
<tr>
<td><code>git checkout develop</code></td>
<td>Checkout the develop branch in the local workspace</td>
</tr>
<tr>
<td><code>git checkout -b my_branch</code></td>
<td>Create a new branch in the workspace with the name <code>my_branch</code></td>
</tr>
<tr>
<td><code>git remote add your_origin yourgit.com/repo.git</code></td>
<td>Add your remote origin to the workspace</td>
</tr>
</tbody>
</table>
Focus on the RAN CI

It is currently the most advanced CI process

But the principles are the same for the CN CI
Where does it start? The GITLAB repository

All CI scripts / files are located here

CI Status Notification

To contribute, reach us at contact@openairinterface.org
A Push vs a Merge Request

• RAN CI jobs will be automatically triggered when:

• A push to develop branch occurs
  • Power users like myself (on the previous slide) do a straight push
  • When a (several) merge request(s) is (are) merged into develop branch

• A Merge Request to develop branch is opened, edited, updated
  • At creation or after edition (for example, assigning someone)
  • Mainly when you are pushing again to the source branch
WIP or not WIP? That is the question

• **WIP**: Work In Progress
• When creating a Merge Request, you can tag it as **WIP**
  • WIP Merge Requests **WILL NOT** trigger CI jobs
• For CI to process, the Merge Request has to be **MERGE-ABLE**.
  • As a matter of fact, it is the first check performed by CI pipeline script
  • At least the MR should be merge-able at Creation (without the WIP tag)
    • If the MR review is taking a long time, it might become un-merge-able with conflicts due to other MR merges

Don’t forget to select box here in order to remove the source branch after the merge!
GitLab CI Notifications

The Pipeline page opens with all the intermediate notifications.

Click on Any to access directly the Jenkins job URL.

Click Here
GitLab CI Notifications on a MR (1)

CI ran 9 times on this MR
GitLab CI Notifications on a MR (2)

CI is using my credentials

Contributor is making a new push
-> CI is starting again

I took another currently opened MR (#432)

OAI Formatting Check

Compilation Warnings are located on the files modified by the MR
You may or may not be responsible for them

Click Here to access Jenkins master job web-page
Notifications on Slack Channel

Public Channel

I made a few mistakes developing CI scripts

You can DM me

Join us at https://openairinterface.slack.com/
Email to the committer

Hi,

Here are attached HTML report files for RAN-CI-develop - Build # 116 - Still Failing!

Regards,

OAI CI Team

Job Summary -- Job: RAN-CI-develop -- Build-ID: 116

Was it a Push or a Merge Request?

All HTML reports
RAN CI Jobs Architecture

Web-hooks are triggered by:
- Merge-Requests
- Pushes to develop branch

Visible from the World Wide Web

We are testing the OAI eNB and the OAI UE now
Master RAN CI Job

Where the CI starts
OS / Platform Flavors

• Currently most of our CI build / testing is done under
  • x86 – Ubuntu 16.04 LTS (xenial)
  • We are building a single variant on a remote CentOS 7.4 server

• I would like to have dedicated eNB master pipeline jobs for
  • x86 – Ubuntu 16.04 LTS (xenial)
  • x86 – Ubuntu 18.04 LTS (bionic)
  • x86 – CentOS 7.4 or 7.6
Clean Containers

• The purpose is to always from a clean sheet

• Currently we are creating “quickly” little Virtual Machines
  • We are using uvt-kvm package for ease of use

• Each Virtual Machine (VM) has to reinstall tools for the variant build
  • Checking if configuration is not missing from the GITLAB repository

• We (CI team) are investigating using Docker instances to speed up process
Deported Processing

Node is a Jenkins terminology == worker

Pipeline executes on

nodea.eurecom.fr

Visible from the World Wide Web

Eurecom private network

---

Web-hooks are triggered by
- Merge-Requests
- Pushes to develop branch

Web-hooks

eNt-CI
Master Job

open5glab.eurecom.fr

---

Guideline checks
Static Code Analysis (cppcheck)
All common variants are built
Tests without any RF HW
- Unit-level simulators
- Full stack simulators
- Ethernet transport

---

Visible from the World Wide Web

---

OAI Continuous Integration Training
Old RAN Master Job Pipeline

Stage where formatting is checked
Currently non blocking

Stages where VM are created and builds started
-> It is sequential so host server won't crash

Build Stage on a remote CENTOS Server (not a VM)

If Any of Build Stages fails, no Testing performed

Not every stage is a GitLab intermediate notification

Slave Jobs

Still the eNB CI job #503!
Slave Jobs
With real radio

All VM-based Radio-less tests on simulators are sequential
⇒ To improve reliability
## Variants Builds

<table>
<thead>
<tr>
<th>Variant Name</th>
<th>Build Options (Default 3GPP-Release is Release 14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>enb-usrp</td>
<td>build_oai --eNB -w USRP --mu</td>
</tr>
<tr>
<td>basic-sim</td>
<td>build_oai --eNB --UE</td>
</tr>
<tr>
<td>phy-sim</td>
<td>build_oai --phy_simulators</td>
</tr>
<tr>
<td>enb-ethernet</td>
<td>build_oai --eNB</td>
</tr>
<tr>
<td>ue-ethernet</td>
<td>build_oai --UE</td>
</tr>
<tr>
<td>cppcheck</td>
<td>cppcheck --enable=warning --force --xml --xml-version=2</td>
</tr>
</tbody>
</table>

- I know “build_oai” has a lot of options
- But we cannot check every option

- If you think a variant is missing and is important to developers / users, let us know
  - We could add a variant in the pipeline
  - The first 2 build variants will soon disappear
What is tested in RAN Master Job?

• Here we are performing **RF-less** testing

• The “physical simulators” unitary tests are testing L1 physical layer

• The “basic simulator” allows to connect an eNB and a single UE through a tunnel
  • Basic connection with Eurecom RIC (RAN Intelligent Controller) (FlexRAN)

• The “L2 simulator” allows to connect an eNB and multiple UEs bypassing the Layer 1 (using the nFAPI)

• The “RF simulator” soon will connect eNB and multiple UEs through a simulated channel
Basic Simulator Testing

Once the connection between all components is complete
• ping UE from EPC VM
• iperf DL from EPC VM to UE
• iperf UL from UE to EPC VM

If the FLEXRAN Controller installation is complete
• Status queries
• Configuration changes
HTML Reporting on builds

Job Summary -- Job: eNb-Cl -- Build-ID: 505

Build Summary

OAI Coding / Formatting Guidelines Check

55 modified files in Merge-Request DO NOT follow OAI rules. 

Details on the job

Ubuntu 16.04 LTS -- Summary

Details on the SCA errors and warnings

OAI Static Code Analysis with CPPCHECK

CPPCHECK found 193 errors and 744 warnings

Details on the job

OAI Build eNB -- USRP option

BUILD was SUCCESSFUL

Details on the job

Click Here to access the Jenkins Job webpage

OAI Build UE -- ETHERNET transport option

BUILD was SUCCESSFUL

Each variant summary for each library

Red Hat (CentOS Linux release 7.4.1708) -- Summary

Red Hat -- OAI Build eNB -- USRP option

BUILD was SUCCESSFUL

More details
HTML Reporting on tests

Job Summary -- Job: RAN-CI-develop -- Build-ID: 101

<table>
<thead>
<tr>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEFRaw</td>
<td>SUCCESSFUL</td>
</tr>
<tr>
<td>Basic Simulator Check</td>
<td>SUCCESSFUL</td>
</tr>
<tr>
<td>Basic Simulator + FlexRan Controller Check</td>
<td>SUCCESSFUL</td>
</tr>
<tr>
<td>L2-NFAPI Simulator Check</td>
<td>SUCCESSFUL</td>
</tr>
<tr>
<td>Physical Simulators Check</td>
<td>SUCCESSFUL</td>
</tr>
</tbody>
</table>

Click Here to access the Jenkins Job webpage

Click Here for more details

Click Here for more details

Click Here for more details
Let see more details on our failing test

<table>
<thead>
<tr>
<th>Log File Name</th>
<th>Command</th>
<th>Status/Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>test_05MHz_etc_etc</td>
<td>ping c 20 192.172.0.2</td>
<td>Packet Loss = 8%</td>
</tr>
<tr>
<td>test_05MHz_etc_etc</td>
<td>perf c 192.172.0.2 -a -L -D -W 125M</td>
<td>Estimate = 15.0 Mbits/sec</td>
</tr>
<tr>
<td>test_05MHz_etc_etc</td>
<td>perf c 192.172.0.2 -a -L -D</td>
<td>Estimate = 10.0 Mbits/sec</td>
</tr>
</tbody>
</table>

Click here to expand the table

Here is the issue
How to get to build log files

Job Summary -- Job: eNb-CI -- Build-ID: 503

Click Here to access the Jenkins Job webpage from the HTML build or test report

Download build or test logs

HTML Reports from Master and Slave Jobs are archived also

Click on the Artifacts tab
VM Build Log Files

2019-12-03

OAI Continuous Integration Training
VM Test Log Files

2019-12-03

OAI Continuous Integration Training
How you can contribute to CI Master Job

• Only a single script is pulled from the develop branch
  • ci-scripts/Jenkinsfile-gitlab
  • It means a modification on this file during a Merge-Request won’t be taken into account unless merged to develop branch
    • Adding / Removing a stage
    • Changing actions within a stage

• All the rest is opened to any contributor:
  • Adding tests to basic-simulator → see ci-scripts/runTestOnVM.sh
  • Adding tests to physical-simulators → see cmake_targets/autotests/test_case_list.xml
  • Adding FLEXRAN commands/queries → see ci-scripts/runTestOnVM.sh
  • Later on we will introduce full L1 and RF simulator stages
How you can contribute to CI Master Job (2)

• Documentation is available:
  • On the Wiki:
    • https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/ci/continuous-integration-home
  • In the repository:
    • https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop/ci-scripts/doc/ci_dev_home.md
Side Branches Testing

• We can have independent CI for a feature branch

• Currently gNB-CI master job is automatically triggered by
  • push or merge request to develop-nr branch

• It has extra-stages on build (gNB and 5G-NR UE)
• Additional Testing on physical simulators
• It is calling the same eNB/UE slave jobs for legacy testing

• The `develop-nbiot` branch also has its own CI job: RAN-CI-develop-nb-iot

• It SHALL be temporary and SHALL be removed once feature branch is merged into develop
Master Branch Maintenance

- The **master** branch has also its own RAN CI job

- Pipeline and Test Suite are specific to **master** branch feature set

- It allows to do merge requests towards **master** branch for patches
  - Since July 2019 (v1.1.0) master release, we did 1 patch
Coverity Scan

Another Static Code Analysis
Synopsys Coverity Scan

• Synopsys is offering to the Open-Source Projects a free limited version of its tool Coverity

• The limitations are:
  • The Analysis is performed on Synopsys Cloud servers
    • Only the Build is done locally on our server
    • Depending on the code size of the project, limited number of analysis per day
    • On the RAN openairinterface project: 1 analysis per day
  • Our analysis is queued and you wait for result:
    • Last Week I submitted an analysis and 225 jobs were in front of mine
    • Still waiting for the result
  • The results are not exportable → Only through the Web Interface
    • History is very limited
Our project on Coverity Scan

- [https://scan.coverity.com/projects/openairinterface5g](https://scan.coverity.com/projects/openairinterface5g)
- Sign up and I will grant you reviewer access

- We have been submitting jobs on each new `develop` branch integration since early May 2019
  - Variant analyzed: `./build_oai -w USRP --eNB --build-coverity-scan`

- Fujitsu is leading the effort to resolve “High Impact” defects
Current Coverity Scan Status

Analysis Metrics
Version: develop-2019-week-23

Jun 11, 2019 1,286,679 0.39
Last Analyzed Lines of Code Analyzed Defect Density

Defect changes since previous build dated Jun 03, 2019
0 Newly detected 0 Eliminated

Defects by status for current build
1,006 Total defects 506 Outstanding 20 Dismissed 480 Fixed

We are fixing defects and trending down!!
Slave eNB/UE CI Jobs

A lot has been already covered by Kentaro SAWADA’s June 2018 training session

I will present the improvements
Modularity

• Currently we have 9 slave jobs (4 in December 2018, 7 in June)
  • Monolithic FDD eNB: Band 7: 5MHz, 10MHz and 20MHz
  • Monolithic FDD eNB: Band 13: 10MHz to test with LTE-M modules
  • Monolithic TDD eNB: Band 40: 5MHz, 10MHz and 20MHz
  • IF4p5/F1 FDD eNB: Band 7: 5MHz, 10MHz and 20MHz
  • IF4p5 TDD eNB: Band 40: 5MHz, 10MHz and 20MHz
  • Monolithic FDD OAI UE: Band 20: Sniffing Commercial eNB
  • Monolithic FDD eNB + UE: Band 7: 5MHz in S1 and noS1 configurations
  • IF4p5 TDD eNB + RRU: Band 38: Multi-RRU Setup
  • Monolithic FDD eNB: Band 13: X2 Handover setup

• Based on the same pipeline and the same python script
  • Only the scenario changes
  • We are now testing also RRC inactivity timers and CDRX
Modularity (2)

• For 5G NR RAN project
  • We have started to run with 2 N310 RF boards
  • Since it is a shared resource with developers at Eurecom,
    • Fully integrated in the merge request process (target branch develop-nr)

• Based on the same python script and same pipeline
Deported Processing

Pipeline executes on

**nodeb.eurecom.fr**
- Common Python
- Common Pipeline
- Test XML is a parameter on slave job

**enbz.eurecom.fr**
- Build eNB with USRP option

**epc4ci.eurecom.fr**
- Nokia LTEBOX EPC
- OpenAirCN EPC

**adbx.eurecom.fr**
- Control UEs

Visible from the World Wide Web

**open5glab.eurecom.fr**
- Any Slave Job

Eurecom private network

Faraday Cage

USRP board
Pipeline Parameters

- Parameters are now part of the slave job configuration
- No more inside the Jenkins pipeline file
- Reason: modularity
  - Same pipeline re-used for several Slave Jobs
Added a Stage in Pipeline

OAI Continuous Integration Training
Lockable Resources

- All slave jobs are launched in parallel
- But they may share HW / SW resources

- Jenkins plugin allows locking a “resource” for the whole pipeline
  - Other pipelines have to wait until unlocked
- Currently a single resource: our CI bench
- Could be split
  - By UE
  - By RF board
  - By EPC
One “Action Class” in Python Script: iperf

- Iperf original implementation was using iperf3
- At Eurecom we are mainly using iperf (2.0)
  - We support UDP in Downlink and Uplink
  - We support TCP in Downlink and Uplink
- We introduced “profiles”: single-ue, balanced, un-balanced
- Checks are performed against the requested throughput in UDP mode
HTML Report Live Generation

• At the end of each test-case inside the XML scenario file
  • Python script appends a row to an HTML report file

• If any test-case in scenario fails after eNB sync
  • Automatic detachment / termination of UE
  • Automatic termination of eNB(s)
    • Analysis of the eNB log file to potentially spot issue

• If successful, during termination of eNB(s)
  • Automatic analysis of eNB log file
Slave Job HTML Report Example

Job Summary -- Job: eNB-CH-IF4p5-FDD-Band7-B210 -- Build-ID:

149

Jump to Final Status

2 UE(s) is(are) connected to ADB bench server
1 CAT-M UE(s) is(are) connected to bench server

Test Summary for xml_files/if4p5_usrp210_band7.xml

Click here to get the Slave job web-page

2019-12-03

OAI Continuous Integration Training

57
Tool Validation

- Recently we have introduced T Tracer support in XML scenario and python script
  - Recording of a RAW file
  - Replaying the extracted messages from the RAW file
  - Tshark recording of the packets between eNB and EPC
Multi-XML scenario Support

• Reason: when failing early in a long scenario, rest of scenario not executed
  • Each Tab is a XML scenario
  • Footer information on eNB/UE characteristics

```
Job Summary -- Job: eNB-Cl-FDD-Band7-B210 -- Build-ID: 901

Test COT S-IT - OAI eNB Monolithic - LTEBOX - FDD - Band7 - USRP B210

Git Repository: https://github.com/eburton/openairinterface5g
Job Trigger: Merge Request
Source Branch: develop_integration_2019_w21
Source Commit: 0c2e32f29b2e0598e092e875e2d3ed3e24c15933c04ed7
Source Commit: 0c2e32f29b2e0598e092e875e2d3ed3e24c15933c04ed7
Source Commit: 0c2e32f29b2e0598e092e875e2d3ed3e24c15933c04ed7
Target Branch: develop

- 2 UE(s) is(are) connected to ADB bench server
- 1 CAT-M UE(s) is(are) connected to bench server

```
How to get to the log files

Job: eNB-CI-FDD-Band7-B210 -- Build-ID: 416

Click Here to access the Jenkins Job webpage from the HTML test report

Click on the Artifacts tab

The most interesting one contains all eNB log files

HTML Report is also archived here
How our CI bench looks like

Inside the Faraday Cage

LTE-M module

USRP Boards
How you can contribute to CI Slave Jobs (1)

• 1 script is pulled from the develop branch
  • ci-scripts/Jenkinsfile-tmp-ran,
  • It means a modification on these files during a Merge-Request won’t be taken into account unless merged to develop branch
    • Adding / Removing a stage on the pipeline
    • Changing actions within a stage

• But modifications on ci-scripts/main.py and one of the XML scenarios under ci-scripts/xml_files are taken into account
  • Adding / Removing test-case(s) inside the XML scenario file(s)
  • Changing the parameters of a test-case inside the XML scenario file(s)
  • Changing the behavior of an “action class” in the python script

• Nonetheless development on these files can be done and check with a template CI job and a push to develop-ci branch
  • https://open5glab.eurecom.fr:8083/jenkins/view/eNB/job/CI-Sandbox-Ran/
How you can contribute to CI Slave Jobs (2)

• All the configuration files used by CI are located under
  • `ci-scripts/conf_files` folder

• CAUTION: there are 4 keywords that are automatically replaced with correct values by CI scripts
  • `CI_ENB_IP_ADDR`
  • `CI_UE_IP_ADDR`
  • `CI_MME_IP_ADDR`
  • `CI_FLEXRAN_CTL_IP_ADDR`
Statistics since June 2018

- eNb-CI / RAN-CI-develop Master job: started 2018, June 12th
  - Ran 506 times + 423 times + 125 times
- gNb-CI Master job: started 2018, June 25th
  - Ran 247 times + 242 times + 215 times
- eNB-CI-FDD-Band7-B210: started 2018, July 30th
  - Ran 419 times + 528 times + 282 times
- eNB-CI-TDD-Band40-B210: started 2018, September 3rd
  - Ran 312 times + 528 times + 282 times
- eNB-CI-IF4p5-FDD/TDD: started 2018, October 16th
  - Ran each 152 times + 528 times + 282 times

As of December 7th, 2018
As of June 20th, 2019
As of November 28th, 2019
More stats

• [https://gitlab.eurecom.fr/oai/openairinterface5g/pipelines/charts](https://gitlab.eurecom.fr/oai/openairinterface5g/pipelines/charts)
CI is Just a Tool for a Merge Decision

- No Automatic Merge even if the CI jobs are passing
- Merge Decision is still a **human** decision
- On consequent contributions:
  - A peer review is performed by one or several Technical Committee member(s)
  - A form is posted in the merge request comments
  - See  [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/oai-policies-home](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/oai-policies-home)

- We do not merge MR by MR:
  - Each week or so, we integrate all approved Merge Request(s) in a branch
  - CI runs on this integration branch
Issues

• LTE-M testing is limited
  • A single module
  • No application on the module (so just a ping)

• "Live" Status of the CI components
  • Public Trello Card
Roadmap for CI/CD and for OAI Code base

• NbIOT fully tested by CI

• All C/C++ files in openairinterface5g repository are meeting OAI coding guidelines (formatting, naming convention, logging...)

NEW
Core Network CI Side
OS / Platform Flavors

• Core Network CI build / testing is done under
  • x86 – Ubuntu 18.04 LTS (bionic) (server edition)

• Generic Kernel is used
"Clean" Containers

- The purpose is to "normally" always from a clean sheet

- But due to some package installation that requires a human reaction, we cannot do it

- Template Virtual Machines are woken up with latest packages already installed
  - Only the git submodules are re-installed
Deported Processing

Web-hooks are triggered by:
• Merge-Requests
• Pushes to develop branch

Visible from the World Wide Web

GUIDELINE CHECKS:
Static Code Analysis (cppcheck)
vNF built in Release mode

Node is a Jenkins terminology == worker

Pipeline executes on

Eurecom private network

nodeb.eurecom.fr

VM
HSS

VM
MME

VM
SPGW-C

VM
SPGW-U

open5glab.eurecom.fr
Main Pipeline

We will soon insert test stages.
Retrieve the job artifacts

NEW

Click to select tab

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>pipeline.log</td>
<td>-</td>
</tr>
<tr>
<td>CN-CI-4G-develop-build-24.tar.bz2</td>
<td>48.7 KB</td>
</tr>
<tr>
<td>capcheck.xml</td>
<td>55.2 KB</td>
</tr>
<tr>
<td>test_results_oai_cni.html</td>
<td>14.9 KB</td>
</tr>
</tbody>
</table>
HTML Report (2)

MME STDOUT Analysis

HSS STDOUT Analysis

SPGWC STDOUT Analysis

SPGWU STDOUT Analysis

Test Summary

Not performed yet. ▲

Details

Details for Compilation Errors and Warnings

End of Build Report – Copyright © 2018 OpenAirInterface. All Rights Reserved.
Long-Run Testing

Improve our component robustness
Why Long-Run testing?

• Whether on RAN or CN CI process, testing time is limited
  • RAN : CI job is around 3 hours, plenty of configurations tested
    • the longest time that an eNB is running: 10 minutes
  • CN : CI job is around 30 minutes without testing yet

• We want to test running the eNB for several hours, the CN for maybe 24 hours
  • A lot of attach / detach procedures with traffic in between
R2LAB in Sophia-Antipolis

• https://r2lab.inria.fr/index.md

• But also COSMOS in New-York City and POWDER in Salt Lake City

• Frame work to run longer scenario in FDD Band7

• Frame work is ready --> create the scenarios and loop in python framework
Continuous Deployment

A vision of our future
Current Status on Eurecom RHEL Cluster

• Manual Deployment of the vNF functions and the RAN functions based on the master branches content

• Very difficult for developers to debug

• More details on my presentation in 2 days:
  • OpenAirInterface deployment in containers at EURECOM 5G data center environment under Red Hat OpenShift
Vision

• Deployment of a Jenkins instance in the OpenShift Cluster

• Duplication of some CI builds / tests in the RHEL8 environment

• Extension of the test-suite in the simulated environment
  • But with pods/containers and no more virtual machines

• Automated builds to the new base images based on the merge request acceptance

• Procedures to properly deploy new pods based on new images
Continuous Delivery

A vision of our future
Why Continuous Delivery?

• Currently what we have is a code base and means to build and use
  • Out-dated wiki pages on some use cases
  • …. 

• Our user community is not necessarily linux-savy or network-savy

• Huge pressure for support on the mailing lists and forums for the power contributors
  • For simple problems --> no added value
What to Deliver?

• Snaps vs Containers vs Deb/RPM files?
• We need to have a reflection on that topic

• Also it should have the most common configurations pre-installed
  • For the RAN --> eNB monolithic vs split architecture, FDD / TDD
  • Separate delivery for eNB and OAI UE?
  • For the EPC --> centralized deployment vs distributed deployment
    • Automatic creation of the virtualized network interfaces????
Thanks for listening

More details on the implementation can be found on the Wiki

https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/ci/continuous-integration-home