



**Linaro  
connect**  
Vancouver 2018

# YVR18-206: Evolution of Continuous Integration Systems for Open Source Projects

Maxim Uvarov



# Introduction

The goal of presentation is to share our experience in bootstrapping CI for open source project.



# What is OpenDataPlane project

The ODP project is an open-source, cross-platform set of APIs for the networking data plane

Project hosted on github:

<https://github.com/Linaro/odp>

ODP is written using C99

We support x86, Arm, MIPS64 and PowerPC platforms



# OpenDataplane is...

The ODP project has been established to produce an open-source, cross-platform set of application programming interfaces (APIs) for the networking data plane. Similar to OpenGL for graphics, but for networking stack.

ODP is the set of APIs describing hardware in software.

It is represented as library both static and dynamic.

Supported ABI (binary) compatibility libodp version to use with different implementations.

Non ABI compat version uses “native types” for more speed optimization.

ODP defines API and does not put any restrictions for implementations.



# OpenDataplane is... (continued)

Known projects similar to ODP are: dpdk, netmap, pf\_ring, libpcap.

Known implementations are: Linux generic, DPDK based, Cavium Thunder-X, Freescale based on DPAA, TI Keystone II, Kalray MPAA.

Current ODP implementations done in linux user space. Where hardware delivers packets directly to user space memory. If ODP API can not be accelerated with hardware, than software realisation is used



# Project was started in 2013

We had simple Makefiles (no autotools). All compilations on local machine. Maintainer did the latest check on his laptop and pushed changes to server.

With that we observed following issues:

- We need tests and some automation for them;



# Use used bash/Makefile scripts (2014-2015)

- Project was switched to autotools (automake, configure.ac);
- Make test, target was implemented to test all ODP API and implementation;
- We started to use homegrown bash/Makefile scripts to set up all environment and run the test.

## **Problems were:**

- Loooong test run.
- Maintainers review code before tests passed;
- Environment on several machines can differ (different gcc versions and etc.)



# Transfer to github and Travis (2015-2016)

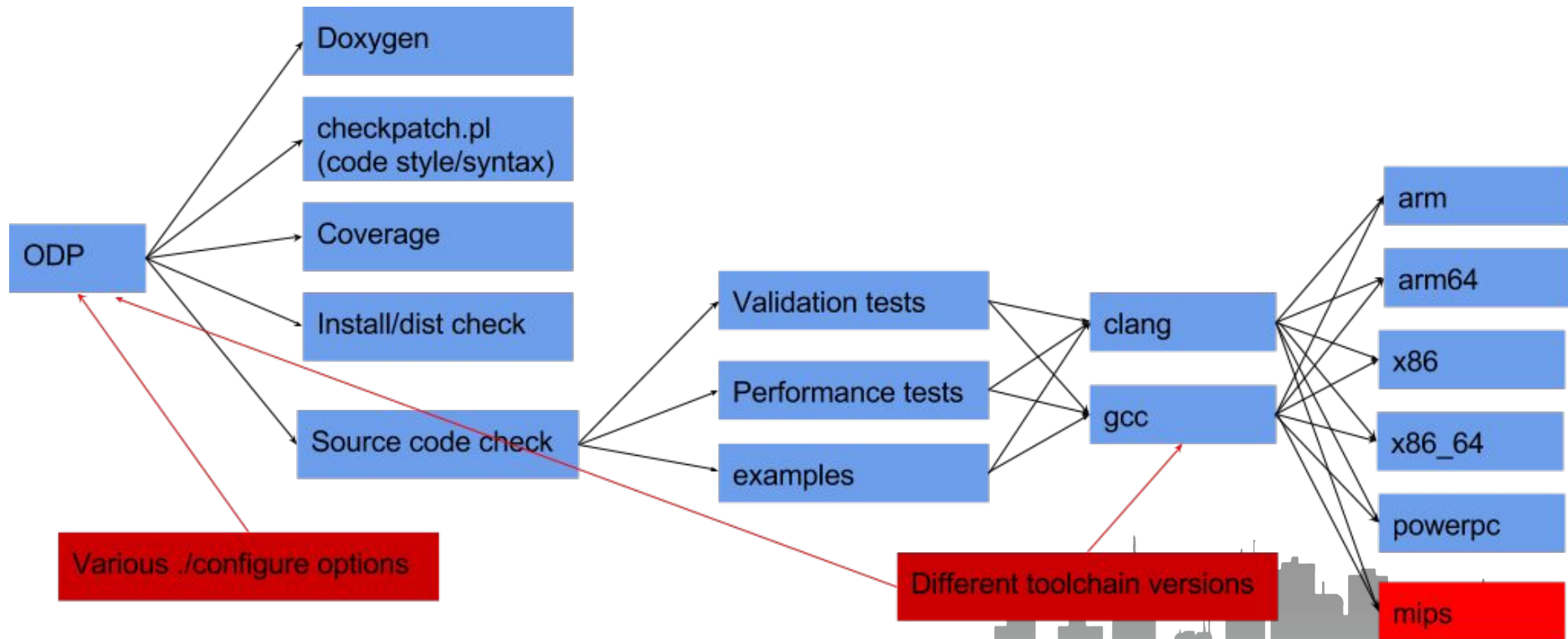
We were happy! We validated:

- Each pull request;
- Each commit in git;
- With our scripts we sent PRs to mailing list and replies from mailing list back to github pull requests;
- We used free plan for open source projects and saved some money on traffic and hardware maintenance;





# We test number of different combinations:



# And in general Github/Travis worked well....



But of course we found some issues...



# One day our Travis test hanged..

- Tests passed locally, but in Travis they don't.
- Root cause that tap device did not pass packets to application which we used for testing;
- Travis updated run image and we could switch back to deprecated image;
- We used that deprecated image for some long time, until some day I updated Ubuntu.
- Default iptable forward policy was changed to drop. It's was easy to debug locally but in Travis it was difficult to debug...
- We still wanted that everyone can compile our project and run tests under his environment.



# Ubuntu also bring us unexpected surprises...

- Ubuntu started to delete mirrors for “old” images. Apt-get some time worked, sometime not.



# We needed to improve everything

- We wanted to run locally exactly the same set up which runs our CI. -  
Now Travis runs docker images;
- We wanted to test on different environments (different Ubuntu, Debian, CentOS versions).
- Tests should cover all code but execution time has to be small;



## Now we use...

Travis which runs bunch of variation of Docker containers.

```
docker run --privileged -i -t
```

```
-v `pwd`:/odp
```

```
-e CC="${CC}" -e LD="${LD}" -e CXX="${CXX}"
```

```
-e CONF="${CONF}" -e DPDK_SHARED="${DPDK_SHARED}"
```

```
${DOCKER_NAMESPACE}/travis-odp-Ing-x86
```

```
/odp/scripts/ci/build_x86_64.sh
```

- Simple .travis.yml file
- Can be run locally with copy-paste from .travis.yml.



# Links to share...

- Project where we build Docker images:  
<https://github.com/Linaro/odp-docker-images>
- .travis.yml and scripts inside main git repo:  
<https://github.com/Linaro/odp>



# Thank you!

