Linaro connect
San Francisco 2017

SFO17-315: OpenDataPlane Testing in Travis

Dmitry Eremin-Solenikov, Cavium
Maxim Uvarov, Linaro

OpenDataPlane.org

LNG Networking

connect.linaro.org

Linaro connect
San Francisco 2017

What is ODP (OpenDataPlane)

- The ODP project is an open-source, cross-platform set of APIs for the networking data plane
- Default implementation is also provided
- ODP is written using C99, with the goal of supporting different hardware platform
  - We support x86, ARM, MIPS64 and PowerPC platforms
- We selected Travis CI as our public CI system

ENGINEERS AND DEVICES WORKING TOGETHER

Linaro connect
San Francisco 2017

What is ODP Continued

- Project hosted on github:

  https://github.com/Linaro/odp
- Provides dynamic and static libraries
- Uses autotools (autoconf, automake).
- Uses Doxygen to generate API documentation

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Linaro connect
San Francisco 2017

ENGINEERS AND DEVICES WORKING TOGETHER

What is Travis CI

- Travis CI is free continuous integration platform, tightly integrated with GitHub hosting system
- It features rich set of features, which we exploited for our development and which we will demonstrate during this talk
  - Scriptable and configurable task descriptions
  - Up to 5 builders running in parallel
  - Networking access, root access, etc
- Easy to start: Just put simple `.travis.yml` file into your project root

Applied Travis CI

- We started with just 2 compilers x 4 build variants matrix
- Then we had to add more build variants...
- … and cross compilation...
- … and few special tasks…
- … and it went on and on…

- Till we had single build taking up to 4 ¼ hours

Linaro connect
San Francisco 2017

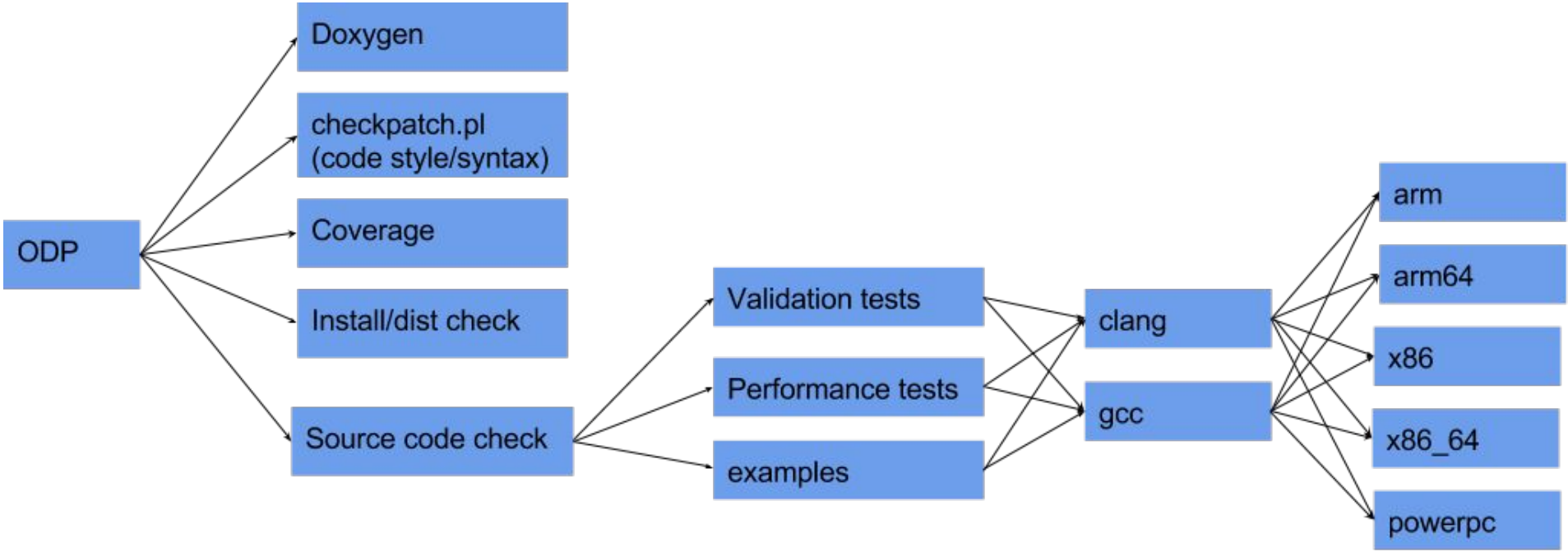
Github and Travis integration

- Travis runs on each git push to the repo once enabled (does not matter which repo is it: main or private fork);
- Validation for each pull request and PR updates;
- Travis cron runs for stable branch;

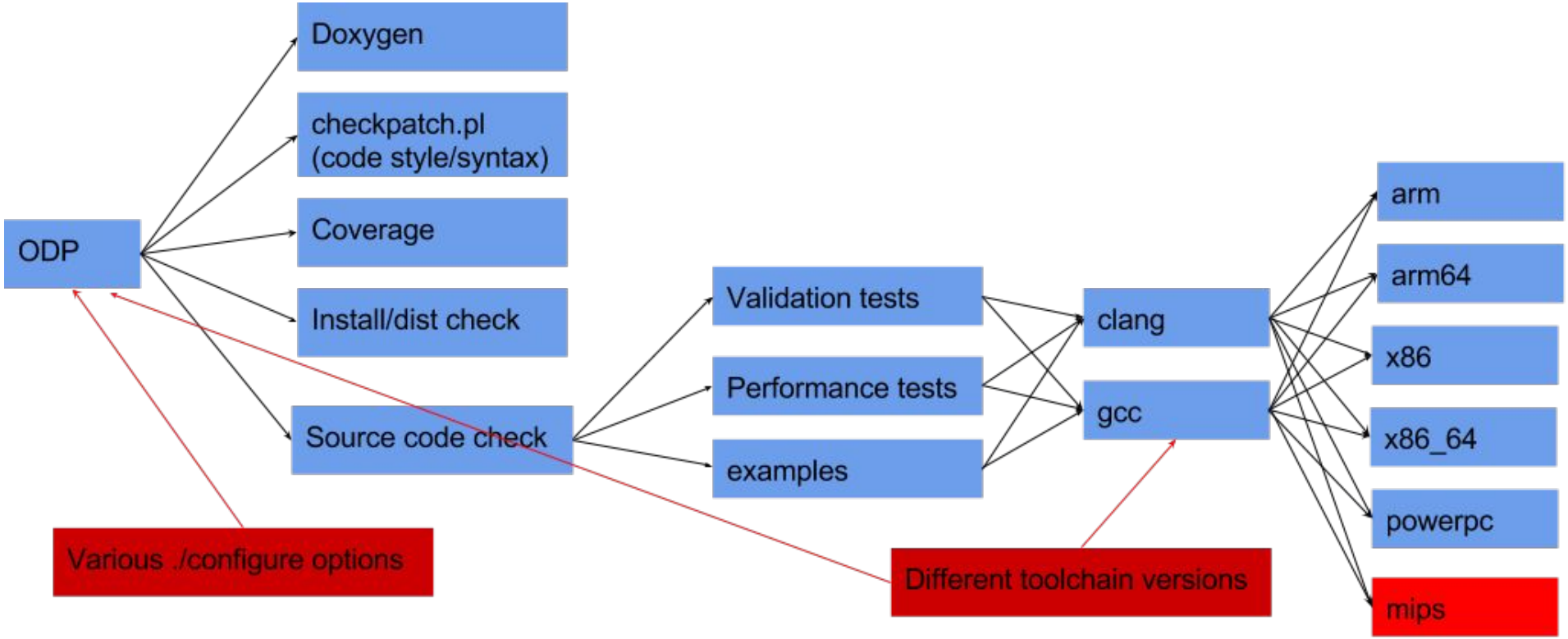
We do not need ‘git bisect’ project. We validate patches before inclusion and know on which commit tests passed!

ENGINEERS
AND DEVICES
WORKING
TOGETHER

ODP testing looks like:



Did we forget something? YES!



Linaro connect
San Francisco 2017

ENGINEERS AND DEVICES WORKING TOGETHER

Build Matrix (now)

✓	# 787.1	Compiler: gcc C	CONF=""	9 min 11 sec
✓	# 787.2	Compiler: clang-3.8 C	CONF=""	10 min 2 sec
✓	# 787.3	Compiler: gcc C	CONF="--disable-abi-compat"	9 min 10 sec
✓	# 787.4	Compiler: clang-3.8 C	CONF="--disable-abi-compat"	9 min 44 sec
✓	# 787.5	Compiler: gcc C	CONF="--enable-schedule-sp"	9 min 11 sec
✓	# 787.6	Compiler: clang-3.8 C	CONF="--enable-schedule-sp"	10 min 17 sec
✓	# 787.7	Compiler: gcc C	CONF="--enable-schedule-iquery"	9 min 22 sec
✓	# 787.8	Compiler: clang-3.8 C	CONF="--enable-schedule-iquery"	9 min 56 sec
✓	# 787.9	Compiler: gcc C	CONF="--enable-dpdk-zero-copy"	9 min 10 sec
✓	# 787.10	Compiler: clang-3.8 C	CONF="--enable-dpdk-zero-copy"	9 min 46 sec
✓	# 787.11	Compiler: aarch64-linux-gnu-gcc C	TEST="aarch64-linux-gnu" CROSS_ARCH="arm64"	2 min 51 sec
✓	# 787.12	Compiler: "clang-3.8 --target=aarch64-linux-gnu" C	TEST="clang-3.8 aarch64-linux-gnu" CROSS_ARCH="arm64"	3 min 33 sec
✓	# 787.13	Compiler: arm-linux-gnueabihf-gcc C	TEST="arm-linux-gnueabihf" CROSS_ARCH="armhf"	2 min 27 sec
✓	# 787.14	Compiler: "clang-3.8 --target=arm-linux-gnueabihf" C	TEST="clang-3.8 arm-linux-gnueabihf" CROSS_ARCH="armhf" CFLAGS="-march=	2 min 31 sec
✓	# 787.15	Compiler: powerpc-linux-gnu-gcc C	TEST="powerpc-linux-gnueabihf" CROSS_ARCH="powerpc"	2 min 50 sec
✓	# 787.16	Compiler: "clang-3.8 --target=powerpc-linux-gnu" C	TEST="clang-3.8 powerpc-linux-gnu" CROSS_ARCH="powerpc"	3 min 18 sec
✓	# 787.17	Compiler: gcc C	TEST=coverage	9 min 45 sec
✓	# 787.18	Compiler: gcc C	TEST=distcheck	12 min 2 sec
✓	# 787.19	Compiler: gcc C	TEST=doxygen	2 min 25 sec
✓	# 787.20	Compiler: gcc C	TEST=checkpatch	1 min 35 sec

Linaro connect
San Francisco 2017

ENGINEERS AND DEVICES WORKING TOGETHER

Gory Details

- Build matrix + jobs.include: two ways to specify build tasks
  - Matrix compilers vs environment
  - Additional jobs added one by one
- Cross-compilation is possible, but required additional efforts (Ubuntu provides cross-toolchains for supported architectures).
- Each additional test is a separate job

Build Matrix in YAML

```
env:
    - CONF=""
    - CONF="--disable-abi-compat"
    - CONF="--enable-schedule-sp"
    - CONF="--enable-schedule-iquery"
    - CONF="--enable-dpdk-zero-copy"
jobs:
    include:
        - stage: test
          compiler: aarch64-linux-gnu-gcc
          env: TEST="aarch64-linux-gnu" CROSS_ARCH="arm64"
          script:
            - ./bootstrap
            - ./configure --prefix=$HOME/odp-install $CROSS --enable-debug
              --disable-test-cpp --enable-test-vald --enable-test-helper --enable-test-perf
--enable-test-perf-proc --enable-test-example
--with-cunit-path=$HOME/cunit-install/$CROSS_ARCH
            - make -j $(nproc)
```

Build caching

- Total build time is 2:20 h
- No need to compile DPDK, netmap, CUnit on each run!

```
cache:
    ccache: true
    pip: true
    directories:
        - dpdk
        - netmap
        - $HOME/cunit-install
```

Code coverage

```
788  1  static inline int move_data_to_tail(odp_packet_hdr_t *pkt_hdr, int segs)
789     {
790             int dst_seg, src_seg;
791             uint32_t len, free_len;
792  1          uint32_t moved = 0;
793
794  6          for (dst_seg = segs - 1; dst_seg >= 0; dst_seg--) {
795  6                  len    = pack_seg_tail(pkt_hdr, dst_seg);
796  6                  moved += len;
797
798  6                  if (len == BASE_LEN)
799                             continue;
800
801  6                  free_len = BASE_LEN - len;
802
803  6                  for (src_seg = dst_seg - 1; src_seg >= 0; src_seg--) {
804  5                          len = fill_seg_tail(pkt_hdr, dst_seg, src_seg,
805                                                 free_len);
806  5                          moved += len;
807
808  5                          if (len == free_len) {
809                                     /* dst seg is full */
810  5                                  break;
811                             }
812
813                             /* src seg is empty */
814                             free_len -= len;
815                     }
816
817  6                  if (moved == pkt_hdr->frame_len)
818  1                          break;
819             }
820
821             /* first segment which have data */
822  1          return dst_seg;
823     }
```

Future steps

- Cross-testing under qemu
- Native ARM test run
- Further time-related optimizations

Linaro connect
San Francisco 2017

Thank You

#SFO17

ODP github page https://github.com/Linaro/odp

Travis CI configuration: https://github.com/Linaro/odp/blob/master/.travis.yml

Travis: https://travis-ci.org/

For further information: www.opendataplane.org