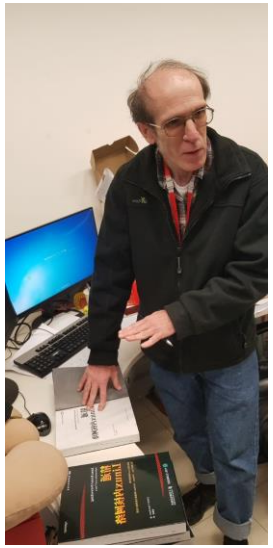
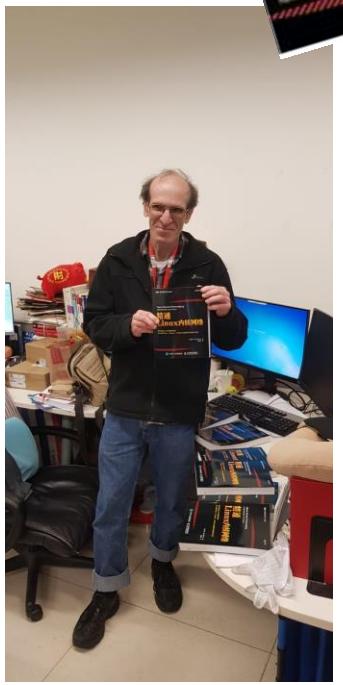
Developing CI/CD for NICs DPDK PMD using DTS

RAMI ROSEN (DECEASED)
SHAHAR BELKAR – TOGA NETWORKS

Rami Rosen (1961 – 2019)

- A friend
- A colleague
- A mentor
- A Linux Kernel expert, the author of "Linux Kernel Networking", Apress, 2014
- https://ramirose.wixsite.com/ramirosen





Agenda

- DTS
  - Overview and Background
  - Basic Setup
  - Reports
  - UNH IOL (Community Lab)
  - Performance Test Lab Dashboard
  - Advantages
- Challenges enabling DTS on a new PMD
- DTS as part of CI/CD
- DTS as a validation environment
- Status & Roadmap

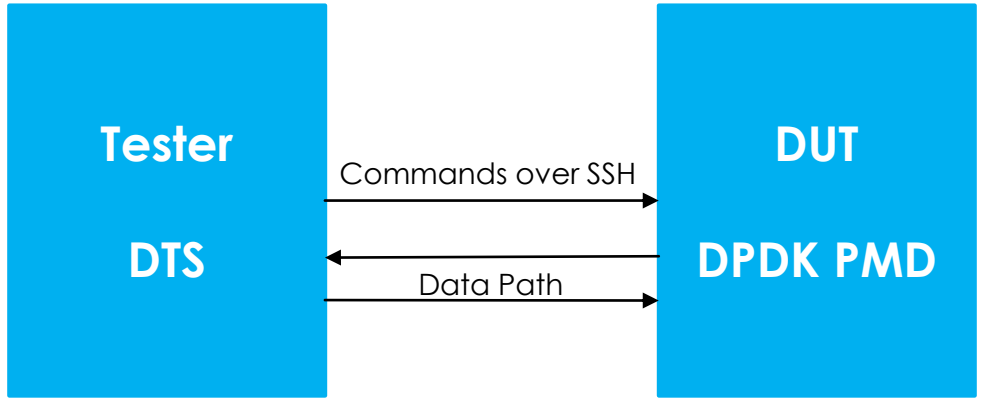
DTS: Overview and background

- DTS - DPDK Test Suite
  - https://doc.dpdk.org/dts/gsg/
  - https://doc.dpdk.org/dts/test_plans/
- DTS was started in 2014 by Intel
  - A Framework written in Python 2.7
  - Consists of *functional tests* and *benchmarking tests*
  - There are over 140 test modules (Called Test Suites in DTS semantics)
  - Uses SW packet generators: scapy, dpdk-pktgen and TRex (by Cisco)
  - Also HW packet generator, IXIA (Using it's TCL API)
- Supports testing of various NICs from:
  - Intel
  - Mellanox
  - Cavium
  - Huawei
- Generates reports in the following forms:
  - RST (ReStructuredText)
  - Excel
  - PDF

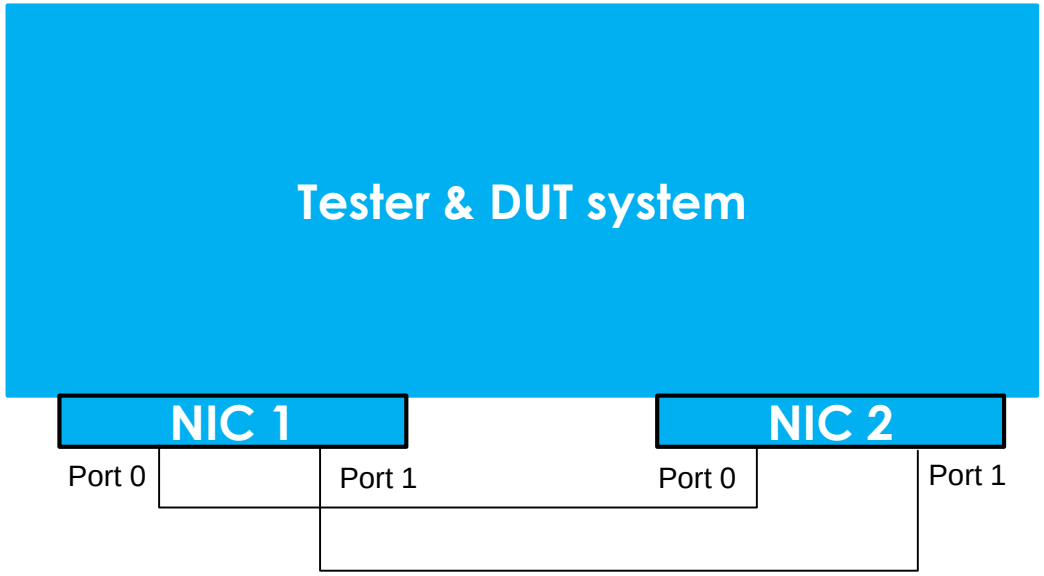
DTS: Basic Setup

- Typically, we use two hosts, connected directly by a cable;
  one is the tester, the second one is the DUT (Device Under Test)
  - You need to set SSH keys so that you can SSH from Tester to DUT without prompting for a password
  - Alternatively, connect an IXIA HW traffic generator instead of the tester
- The DUT has DPDK installed on it, by default under /root/dpdk

(1)



(2)



DTS: Reports (RSS test as an example)

Redirection table Tests

This document provides test plan for benchmarking of Rss reta(Redirection table) updating a Poll Mode Driver (PMD) in userland runtime configurations. The content of Rss Redirection table are not defined following reset of the Memory Configuration registers.
System software must initialize the table prior to enabling multiple receive queues. It can also update the redirection table during run time. Such updates of the table are not synchronized with the arrival time of received packets.

DPDK Version:19.05.0-rc0 TRC. Date:27-03-19 FW: 1.6.2.5

Added by us

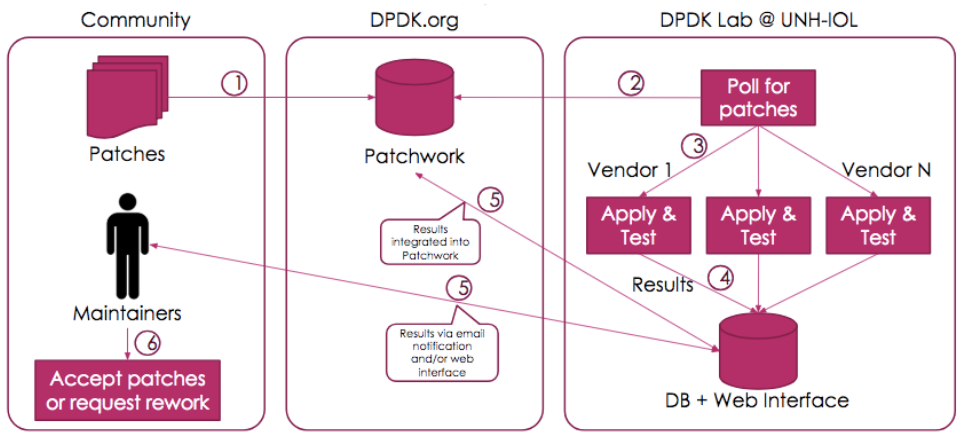
Taken from DTS properties

Test Case: test_pmdrss_reta

packet index	hash value	hash index	queue id	actual queue id	pass
0	0x827a87fa	250	1	1	true
1	0xeb0c4370	112	0	0	true
2	0x2190f621	33	1	1	true
3	0x52ff77d2	210	1	1	true
4	0x152b9464	100	0	0	true
5	0x7c5d50ee	238	1	1	true
6	0x7065cecc	204	0	0	true

DTS: UNH IOL (Community Lab)

- DTS is used by UNH IOL
  - University of New Hampshire Inter-Operability Lab
  - DPDK Performance Test Lab
  - https://www.iol.unh.edu/testing/hpc/dpdk
  - https://lab.dpdk.org/results/dashboard/



(*)

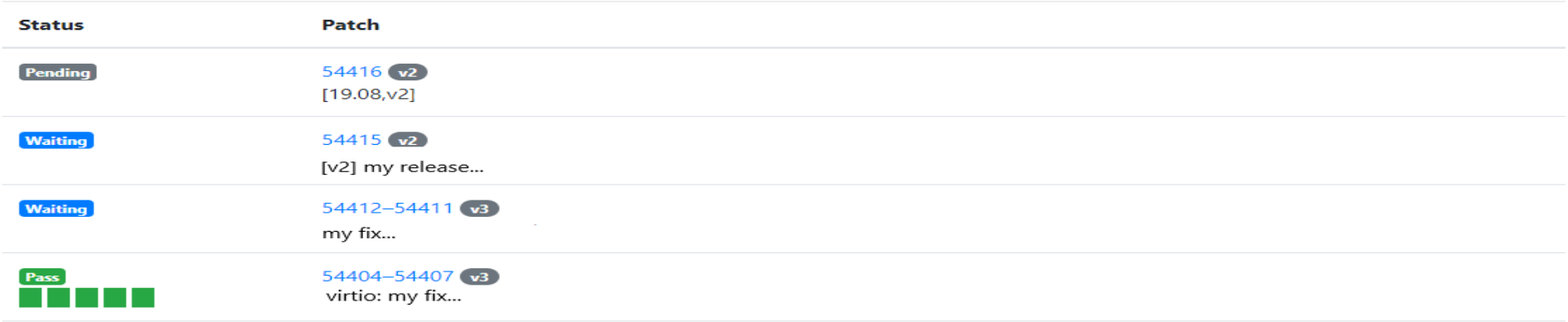
- For each patch which is sent over the dpdk-dev mailing list, an automated performance regression testing is launched
- The degradation/boost in performance (in percentage), relative to a known baseline, is published on the public site of the lab
- Currently this is done on a variety of Intel & Mellanox NICs

University of New Hampshire InterOperability Laboratory

*Taken from DPDK website - http://core.dpdk.org/lab/

DTS: Performance Test Lab Dashboard

- Dashboard

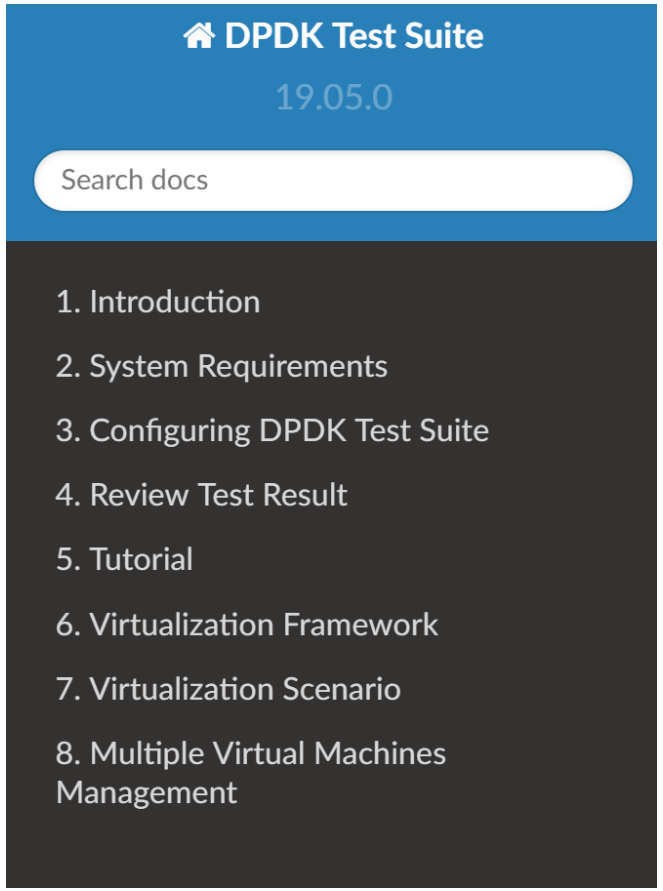


- Performance report

Result	frame_size (bytes)	txd/rxd (descriptors)	throughput Difference (Mpps)
PASS	64	256	0.68600
PASS	128	256	0.26300
PASS	256	256	-0.05100
PASS	512	256	0.03100
PASS	1024	256	0.00400
PASS	1518	256	-0.02000

DTS: Advantages

- Doesn’t require to re-invent the wheel
- Open source (share knowledge with community)
- Updated before each DPDK formal release
- Rich documentation and user guides
- Easy to use and friendly environment



DTS: Advantages – Cont.

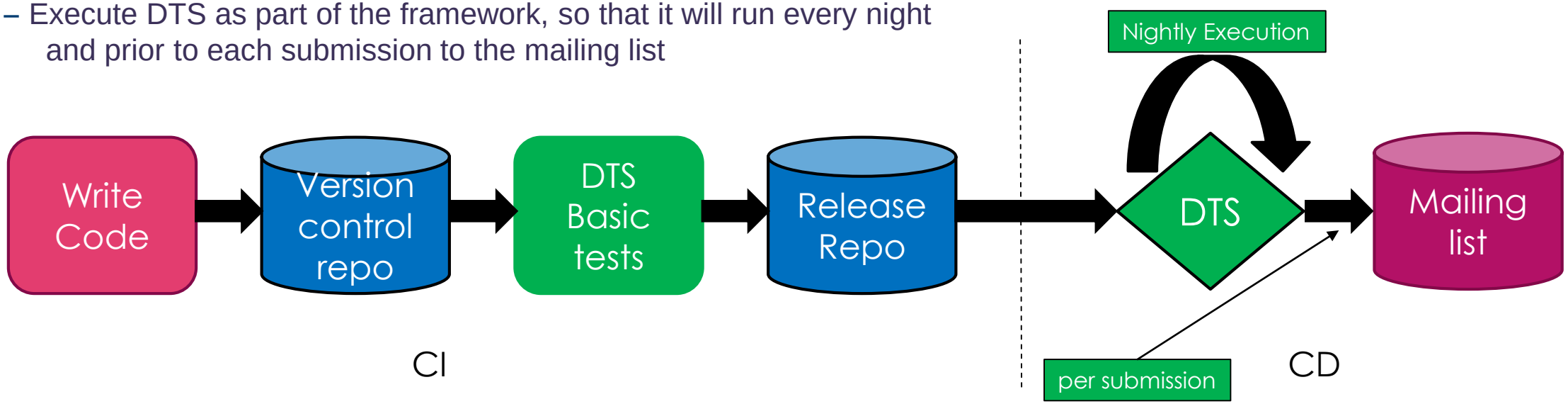
- Adopted by the major vendors
- Compatible with different NICs and platforms
- Performance measurements
- Supports virtualization (VM configuration, different hypervisors supported etc)
- Aligns the commits base line and improves the quality of open source patch
- Enables a standardized and fully automated environment for CI/CD

Challenges Enabling DTS on a New PMD (IN200)

- Some of the tests are specific to intel NICs (i40e, FVL, Niantic, etc)
  - E.g. the ddp_test (Dynamic Device Personalization), is Intel FVL specific
  - In some cases, the test module name includes it
- Some of the tests required adaptations in the DTS code
  - checksum_offload
  - l2fw
  - tso
- Some tests required adaptation in the PMD
  - We found out that in order for the jumbo frame test to work, we need to set DEV_RX_OFFLOAD_JUMBO_FRAME | DEV_RX_OFFLOAD_SCATTER for RX,
    and DEV_TX_OFFLOAD_MULTI_SEGS for TX in the hinic_dev_infos_get() method
  - Vhost PMD Xstats test required specific values and naming

DTS as part of CI/CD

- Why?
  - Reduce issues which can occur when submitting new patches
  - Submit higher quality patches and avoid later fix patches
  - Track any performance degradation in advance
- How?
  - Update the DPDK repo on a daily basis against DPDK master. Make sure there are no new DTS failures
  - Run a small portion of DTS for acceptance per developer commit to the local repo
  - Execute DTS as part of the framework, so that it will run every night and prior to each submission to the mailing list



DTS as a Validation Environment (SW/FW/HW)

- Robust environment
- Set a base line for higher quality PMDs
- Allows full flexibility
- Assist each other in creating the base line, fixing bugs and generalizing test cases
- Requires changes in the environment:
  - Clean up after failure/crash (catch and clean)
  - Prevent parallel executions
  - Create separation for specific vendor feature (e.g. DDP)
  - Unified test framework (already proposed by Intel)
    - Task assignment
    - Resource management
    - Case management
    - Reports
    - Guest agent

Status & Roadmap

- Triggering a few DTS tests after each commit (basic acceptance)
- Triggering the DTS before each commit to the mailing list
  - Many TestSuites are passing, where some required adaptations due to the specific NIC's behavior
  - Committed the patches to the mailing list, targeting DPDK 19.08, after relevant DTS TestSuites passed
- Future work
  - Create a smart acceptance per commit (with time boundaries – up to 10 min)
  - Automatically deploy the environment to any matching setup (vs. dedicated)
  - Widen the coverage (OS types and platforms)
  - Automatically run full DTS test suite on a nightly basis
  - Encourage the DPDK community to run DTS as part of their CI/CD
  - Upgrade to Python 3, as Python 2.7 is end of life in half a year (January 1, 2020)

Questions?

Shahar Belkar

Backup