DPDK

DATA PLANE DEVELOPMENT KIT

Release Notes

Release 18.11.0

November 27, 2018

CONTENTS

CHAPTER

ONE

DPDK RELEASE 18.11

1.1 New Features

- **Added support for using externally allocated memory in DPDK.**

  DPDK has added support for creating new `rte_malloc` heaps referencing memory that was created outside of DPDK's own page allocator, and using that memory natively with any other DPDK library or data structure.

- **Added check for ensuring allocated memory is addressable by devices.**

  Some devices can have addressing limitations so a new function, `rte_mem_check_dma_mask()`, has been added for checking that allocated memory is not out of the device range. Since memory can now be allocated dynamically after initialization, a DMA mask is stored and any new allocated memory will be checked against it and rejected if it is out of range. If more than one device has addressing limitations, the DMA mask is the more restrictive one.

- **Updated the C11 memory model version of the ring library.**

  Added changes to decrease latency for architectures using the C11 memory model version of the ring library.

  On Cavium ThunderX2 platform, the changes decreased latency by 27-29% and 3-15% for MPMC and SPSC cases respectively (with 2 lcores). The real improvements may vary with the number of contending lcores and the size of the ring.

- **Added hot-unplug handle mechanism.**

  Added `rte_dev_hotplug_handle_enable()` and `rte_dev_hotplug_handle_disable()` for enabling or disabling the hotplug handle mechanism.

- **Added support for device multi-process hotplug.**

  Added support for hotplug and hot-unplug in a multiprocessing scenario. Any ethdev devices created in the primary process will be regarded as shared and will be available for all DPDK processes. Synchronization between processes will be done using DPDK IPC.

- **Added new Flow API actions to rewrite fields in packet headers.**

  Added new Flow API actions to:

  – Modify source and destination IP addresses in the outermost IPv4/IPv6 headers.

  – Modify source and destination port numbers in the outermost TCP/UDP headers.

- **Added new Flow API action to swap MAC addresses in Ethernet header.**

  Added new Flow API action to swap the source and destination MAC addresses in the outermost Ethernet header.

- **Add support to offload more flow match and actions for CXGBE PMD.**

  The Flow API support has been enhanced for the CXGBE Poll Mode Driver to offload:

  – Match items: destination MAC address.

  – Action items: push/pop/rewrite vlan header, rewrite IP addresses in outermost IPv4/IPv6 header, rewrite port numbers in outermost TCP/UDP header, swap MAC addresses in outermost Ethernet header.

- **Added a devarg to use the latest supported vector path in i40e.**

  A new devarg `use-latest-supported-vec` was introduced to allow users to choose the latest vector path that the platform supported. For example, users can use AVX2 vector path on BDW/HSW to get better performance.

- **Added support for SR-IOV in netvsc PMD.**

  The `netvsc` poll mode driver now supports the Accelerated Networking SR-IOV option in Hyper-V and Azure. This is an alternative to the previous vdev_netvsc, tap, and failsafe drivers combination.

- **Added a new net driver for Marvell Armada 3k device.**

  Added the new `mvneta` net driver for Marvell Armada 3k device. See the `../nics/mvneta` NIC guide for more details on this new driver.

- **Added NXP ENETC PMD.**

  Added the new enetc driver for the NXP enetc platform. See the `../nics/enetc` NIC driver guide for more details on this new driver.

- **Added Ethernet poll mode driver for Aquantia aQtion family of 10G devices.**

  Added the new `atlantic` ethernet poll mode driver for Aquantia XGBE devices. See the `../nics/atlantic` NIC driver guide for more details on this driver.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  – Improved security of PMD to prevent the NIC from getting stuck when the application misbehaves.

  – Reworked flow engine to supported e-switch flow rules (transfer attribute).

  – Added support for header re-write(L2-L4), VXLAN encap/decap, count, match on TCP flags and multiple flow groups with e-switch flow rules.

  – Added support for match on metadata, VXLAN and MPLS encap/decap with flow rules.

  – Added support for `RTE_ETH_DEV_CLOSE_REMOVE` flag to provide better support for representors.

  – Added support for meson build.

  – Fixed build issue with PPC.

  - Added support for BlueField VF.
  - Added support for externally allocated static memory for DMA.

- **Updated Solarflare network PMD.**

  Updated the sfc_efx driver including the following changes:

  - Added support for Rx scatter in EF10 datapath implementation.
  - Added support for Rx descriptor status API in EF10 datapath implementation.
  - Added support for TSO in EF10 datapath implementation.
  - Added support for Tx descriptor status API in EF10 (ef10 and ef10_simple) datapaths implementation.

- **Updated the enic driver.**

  - Added AVX2-based vectorized Rx handler.
  - Added VLAN and checksum offloads to the simple Tx handler.
  - Added the “count” flow action.
  - Enabled the virtual address IOVA mode.

- **Updated the failsafe driver.**

  Updated the failsafe driver including the following changes:

  - Added support for Rx and Tx queues start and stop.
  - Added support for Rx and Tx queues deferred start.
  - Added support for runtime Rx and Tx queues setup.
  - Added support multicast MAC address set.

- **Added a devarg to use a PCAP interface physical MAC address.**

  A new devarg `phy_mac` was introduced to allow users to use the physical MAC address of the selected PCAP interface.

- **Added TAP Rx/Tx queues sharing with a secondary process.**

  Added support to allow a secondary process to attach a TAP device created in the primary process, probe the queues, and process Rx/Tx in a secondary process.

- **Added classification and metering support to SoftNIC PMD.**

  Added support for flow classification (rte_flow API), and metering and policing (rte_mtr API) to the SoftNIC PMD.

- **Added Crypto support to the Softnic PMD.**

  The Softnic is now capable of processing symmetric crypto workloads such as cipher, cipher-authentication chaining, and AEAD encryption and decryption. This is achieved by calling DPDK Cryptodev APIs.

- **Added cryptodev port to port library.**

  Cryptodev port is a shim layer in the port library that interacts with DPDK Cryptodev PMDs including burst enqueuing and dequeuing crypto operations.

- **Added symmetric cryptographic actions to the pipeline library.**

  In the pipeline library support was added for symmetric crypto action parsing and an action handler was implemented. The action allows automatic preparation of the crypto operation with the rules specified such as algorithm, key, and IV, etc. for the cryptodev port to process.

- **Updated the AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for the AES-GCM algorithm.

- **Added NXP CAAM JR PMD.**

  Added the new caam job ring driver for NXP platforms. See the `../cryptodevs/caam_jr` guide for more details on this new driver.

- **Added support for GEN3 devices to Intel QAT driver.**

  Added support for the third generation of Intel QuickAssist devices.

- **Updated the QAT PMD.**

  The QAT PMD was updated with additional support for:

  – The AES-CMAC algorithm.

- **Added support for Dynamic Huffman Encoding to Intel QAT comp PMD.**

  The Intel QuickAssist (QAT) compression PMD has been updated with support for Dynamic Huffman Encoding for the Deflate algorithm.

- **Added Event Ethernet Tx Adapter.**

  Added event ethernet Tx adapter library that provides configuration and data path APIs for the ethernet transmit stage of an event driven packet processing application. These APIs abstract the implementation of the transmit stage and allow the application to use eventdev PMD support or a common implementation.

- **Added Distributed Software Eventdev PMD.**

  Added the new Distributed Software Event Device (DSW), which is a pure-software eventdev driver distributing the work of scheduling among all eventdev ports and the lcores using them. DSW, compared to the SW eventdev PMD, sacrifices load balancing performance to gain better event scheduling throughput and scalability.

- **Added extendable bucket feature to hash library (rte_hash).**

  This new "extendable bucket" feature provides 100% insertion guarantee to the capacity specified by the user by extending hash table with extra buckets when needed to accommodate the unlikely event of intensive hash collisions. In addition, the internal hashing algorithm was changed to use partial-key hashing to improve memory efficiency and lookup performance.

- **Added lock free reader/writer concurrency to hash library (rte_hash).**

  Lock free reader/writer concurrency prevents the readers from getting blocked due to a preempted writer thread. This allows the hash library to be used in scenarios where the writer thread runs on the control plane.

- **Added Traffic Pattern Aware Power Control Library.**

Added an experimental library that extends the Power Library and provides empty_poll APIs. This feature measures how many times empty_polls are executed per core and uses the number of empty polls as a hint for system power management.

See the `../prog_guide/power_man` section of the DPDK Programmers Guide document for more information.

- **Added JSON power policy interface for containers.**

  Extended the Power Library and vm_power_manager sample app to allow power policies to be submitted via a FIFO using JSON formatted strings. Previously limited to Virtual Machines, this feature extends power policy functionality to containers and host applications that need to have their cores frequency controlled based on the rules contained in the policy.

- **Added Telemetry API.**

  Added a new telemetry API which allows applications to transparently expose their telemetry in JSON via a UNIX socket. The JSON can be consumed by any Service Assurance agent, such as CollectD.

- **Updated KNI kernel module, rte_kni library, and KNI sample application.**

  Updated the KNI kernel module with a new kernel module parameter, `carrier=[on|off]` to allow the user to control the default carrier state of the KNI kernel network interfaces. The default carrier state is now set to `off`, so the interfaces cannot be used until the carrier state is set to `on` via `rte_kni_update_link` or by writing `1` to `/sys/devices/virtual/net/<iface>/carrier`. In previous versions the default carrier state was left undefined. See `../prog_guide/kernel_nic_interface` for more information.

  Also added the new API function `rte_kni_update_link()` to allow the user to set the carrier state of the KNI kernel network interface.

  Also added a new command line flag `-m` to the KNI sample application to monitor and automatically reflect the physical NIC carrier state to the KNI kernel network interface with the new `rte_kni_update_link()` API. See `../sample_app_ug/kernel_nic_interface` for more information.

- **Added ability to switch queue deferred start flag on testpmd app.**

  Added a console command to testpmd app, giving ability to switch `rx_deferred_start` or `tx_deferred_start` flag of the specified queue of the specified port. The port must be stopped before the command call in order to reconfigure queues.

- **Add a new sample application for vDPA.**

  The vdpa sample application creates vhost-user sockets by using the vDPA backend. vDPA stands for vhost Data Path Acceleration which utilizes virtio ring compatible devices to serve virtio driver directly to enable datapath acceleration. As vDPA driver can help to set up vhost datapath, this application doesn't need to launch dedicated worker threads for vhost enqueue/dequeue operations.

- **Added cryptodev FIPS validation example application.**

  Added an example application to parse and perform symmetric cryptography computation to the NIST Cryptographic Algorithm Validation Program (CAVP) test vectors.

- **Allow unit test binary to take parameters from the environment.**

  The unit test "test", or "dpdk-test", binary is often called from scripts, which can make passing additional parameters, such as a coremask, difficult. Support has been added to the application to allow it to take additional command-line parameter values from the `DPDK_TEST_PARAMS` environment variable to make this application easier to use.

1.2 API Changes

- eal: `rte_memseg_list` structure now has an additional flag indicating whether the memseg list is externally allocated. This will have implications for any users of memseg-walk-related functions, as they will now have to skip externally allocated segments in most cases if the intent is to only iterate over internal DPDK memory.

  In addition the `socket_id` parameter across the entire DPDK has gained additional meaning, as some socket ID's will now be representing externally allocated memory. No changes will be required for existing code as backwards compatibility will be kept, and those who do not use this feature will not see these extra socket ID's. Any new API's must not check socket ID parameters themselves, and must instead leave it to the memory subsystem to decide whether socket ID is a valid one.

- eal: The following devargs functions, which were deprecated in 18.05, were removed in 18.11: `rte_eal_parse_devargs_str()`, `rte_eal_devargs_add()`, `rte_eal_devargs_type_count()`, and `rte_eal_devargs_dump()`.

- eal: The parameters of the function `rte_devargs_remove()` have changed from bus and device names to `struct rte_devargs`.

- eal: The deprecated functions attach/detach were removed in 18.11. `rte_eal_dev_attach` can be replaced by `rte_dev_probe` or `rte_eal_hotplug_add`. `rte_eal_dev_detach` can be replaced by `rte_dev_remove` or `rte_eal_hotplug_remove`.

- eal: The scope of `rte_eal_hotplug_add()`/`rte_dev_probe()` and `rte_eal_hotplug_remove()`/`rte_dev_remove()` has been extended. In the multi-process model, they will guarantee that the device is attached or detached on all processes.

- mbuf: The `__rte_mbuf_raw_free()` and `__rte_pktmbuf_prefree_seg()` functions were deprecated since 17.05 and are replaced by `rte_mbuf_raw_free()` and `rte_pktmbuf_prefree_seg()`.

- ethdev: The deprecated functions attach/detach were removed in 18.11. `rte_eth_dev_attach()` can be replaced by `RTE_ETH_FOREACH_MATCHING_DEV` and `rte_dev_probe()` or `rte_eal_hotplug_add()`. `rte_eth_dev_detach()` can be replaced by `rte_dev_remove()` or `rte_eal_hotplug_remove()`.

- ethdev: A call to `rte_eth_dev_release_port()` has been added in `rte_eth_dev_close()`. As a consequence, a closed port is freed and seen as invalid because of its state `RTE_ETH_DEV_UNUSED`. This new behavior is enabled per driver for a migration period.

- A new device flag, `RTE_ETH_DEV_NOLIVE_MAC_ADDR`, changes the order of actions inside `rte_eth_dev_start()` regarding MAC set. Some NICs do not support MAC

changes once the port has started and with this new device flag the MAC can be properly configured in any case. This is particularly important for bonding.

- The default behavior of CRC strip offload has changed in this release. Without any specific Rx offload flag, default behavior by a PMD is now to strip CRC. `DEV_RX_OFFLOAD_CRC_STRIP` offload flag has been removed. To request keeping CRC, application should set `DEV_RX_OFFLOAD_KEEP_CRC` Rx offload.
- eventdev: The type of the second parameter to `rte_event_eth_rx_adapter_caps_get()` has been changed from uint8_t to uint16_t.
- kni: By default, interface carrier status is `off` which means there won't be any traffic. It can be set to `on` via `rte_kni_update_link()` API or via `sysfs` interface: `echo 1 > /sys/class/net/vEth0/carrier`.

  Note interface should be `up` to be able to read/write sysfs interface. When KNI sample application is used, `-m` parameter can be used to automatically update the carrier status for the interface.
- kni: When ethtool support is enabled (`CONFIG_RTE_KNI_KMOD_ETHTOOL=y`) ethtool commands `ETHTOOL_GSET & ETHTOOL_SSET` are no longer supported for kernels that have `ETHTOOL_GLINKSETTINGS & ETHTOOL_SLINKSETTINGS` support. This means `ethtool "-a|--show-pause", "-s|--change"` won't work, and `ethtool <iface>` output will have less information.

1.3 ABI Changes

- eal: added `legacy_mem` and `single_file_segments` values to `rte_config` structure on account of improving DPDK usability when using either `--legacy-mem` or `--single-file-segments` flags.
- eal: EAL library ABI version was changed due to previously announced work on supporting external memory in DPDK:
  - Structure `rte_memseg_list` now has a new field indicating length of memory addressed by the segment list
  - Structure `rte_memseg_list` now has a new flag indicating whether the memseg list refers to external memory
  - Structure `rte_malloc_heap` now has a new field indicating socket ID the malloc heap belongs to
  - Structure `rte_mem_config` has had its `malloc_heaps` array resized from `RTE_MAX_NUMA_NODES` to `RTE_MAX_HEAPS` value
  - Structure `rte_malloc_heap` now has a `heap_name` member
  - Structure `rte_eal_memconfig` has been extended to contain next socket ID for externally allocated segments
- eal: Added `dma_maskbits` to `rte_mem_config` for keeping the most restrictive DMA mask based on the devices addressing limitations.
- eal: The structure `rte_device` has a new field to reference a `rte_bus`. It thus changes the size of the `struct rte_device` and the inherited device structures of all buses.

1.4 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
     librte_acl.so.2
     librte_bbdev.so.1
     librte_bitratestats.so.2
     librte_bpf.so.1
   + librte_bus_dpaa.so.2
   + librte_bus_fslmc.so.2
   + librte_bus_ifpga.so.2
   + librte_bus_pci.so.2
   + librte_bus_vdev.so.2
   + librte_bus_vmbus.so.2
     librte_cfgfile.so.2
     librte_cmdline.so.2
     librte_compressdev.so.1
     librte_cryptodev.so.5
     librte_distributor.so.1
   + librte_eal.so.9
     librte_efd.so.1
   + librte_ethdev.so.11
   + librte_eventdev.so.6
     librte_flow_classify.so.1
     librte_gro.so.1
     librte_gso.so.1
     librte_hash.so.2
     librte_ip_frag.so.1
     librte_jobstats.so.1
     librte_kni.so.2
     librte_kvargs.so.1
     librte_latencystats.so.1
     librte_lpm.so.2
     librte_mbuf.so.4
     librte_member.so.1
     librte_mempool.so.5
     librte_meter.so.2
     librte_metrics.so.1
     librte_net.so.1
     librte_pci.so.1
     librte_pdump.so.2
     librte_pipeline.so.3
     librte_pmd_bnxt.so.2
     librte_pmd_bond.so.2
     librte_pmd_i40e.so.2
     librte_pmd_ixgbe.so.2
     librte_pmd_dpaa2_qdma.so.1
     librte_pmd_ring.so.2
     librte_pmd_softnic.so.1
     librte_pmd_vhost.so.2
     librte_port.so.3
     librte_power.so.1
     librte_rawdev.so.1
     librte_reorder.so.1
     librte_ring.so.2
     librte_sched.so.1
     librte_security.so.1
     librte_table.so.3
     librte_timer.so.1
     librte_vhost.so.4
```

1.5 Known Issues

- When using SR-IOV (VF) support with netvsc PMD and the Mellanox mlx5 bifurcated driver the Linux netvsc device must be brought up before the netvsc device is unbound and passed to the DPDK.
- IBM Power8 is not supported in this release of DPDK. IBM Power9 is supported.
- `AVX-512` support has been disabled for `GCC` builds [1] because of a crash [2]. This can affect `native` machine type build targets on the platforms that support `AVX512F` like `Intel Skylake` processors, and can cause a possible performance drop. The immediate workaround is to use `clang` compiler on these platforms. The issue has been identified as a GCC defect and reported to the GCC community [3]. Further actions will be taken based on the GCC defect result.
  - [1]: Commit 8d07c82b239f ("mk: disable gcc AVX512F support")
  - [2]: https://bugs.dpdk.org/show_bug.cgi?id=97
  - [3]: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=88096

1.6 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations
  - CPU
    - Intel(R) Atom(TM) CPU C3758 @ 2.20GHz
    - Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
    - Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
    - Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz
  - OS:
    - CentOS 7.5
    - Fedora 25
    - Fedora 28
    - FreeBSD 11.2
    - Red Hat Enterprise Linux Server release 7.5
    - Open SUSE 15
    - Wind River Linux 8
    - Ubuntu 14.04
    - Ubuntu 16.04
    - Ubuntu 16.10

        * Ubuntu 17.10
        * Ubuntu 18.04
    - NICs:
        * Intel(R) 82599ES 10 Gigabit Ethernet Controller
            · Firmware version: 0x61bf0001
            · Device id (pf/vf): 8086:10fb / 8086:10ed
            · Driver version: 5.2.3 (ixgbe)
        * Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
            · Firmware version: 0x800003e7
            · Device id (pf/vf): 8086:15ad / 8086:15a8
            · Driver version: 4.4.6 (ixgbe)
        * Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
            · Firmware version: 6.01 0x80003221
            · Device id (pf/vf): 8086:1572 / 8086:154c
            · Driver version: 2.4.6 (i40e)
        * Intel(R) Corporation Ethernet Connection X722 for 10GbE SFP+ (4x10G)
            · Firmware version: 3.33 0x80000fd5 0.0.0
            · Device id (pf/vf): 8086:37d0 / 8086:37cd
            · Driver version: 2.4.6 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
            · Firmware version: 6.01 0x80003221
            · Device id (pf/vf): 8086:158b / 8086:154c
            · Driver version: 2.4.6 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
            · Firmware version: 6.01 0x8000321c
            · Device id (pf/vf): 8086:1583 / 8086:154c
            · Driver version: 2.4.6 (i40e)
        * Intel(R) Corporation I350 Gigabit Network Connection
            · Firmware version: 1.63, 0x80000dda
            · Device id (pf/vf): 8086:1521 / 8086:1520
            · Driver version: 5.4.0-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
    - CPU:
        * Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz

    - Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
    - Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
    - Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
  - OS:
    - Red Hat Enterprise Linux Server release 7.6 (Maipo)
    - Red Hat Enterprise Linux Server release 7.5 (Maipo)
    - Red Hat Enterprise Linux Server release 7.4 (Maipo)
    - Red Hat Enterprise Linux Server release 7.3 (Maipo)
    - Red Hat Enterprise Linux Server release 7.2 (Maipo)
    - Ubuntu 18.10
    - Ubuntu 18.04
    - Ubuntu 17.10
    - Ubuntu 16.04
    - SUSE Linux Enterprise Server 15
  - MLNX_OFED: 4.4-2.0.1.0
  - MLNX_OFED: 4.5-0.3.1.0
  - NICs:
    - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1007
      - Firmware version: 2.42.5000
    - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
      - Host interface: PCI Express 3.0 x8

      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.23.8022 and above
    * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013

            - Firmware version: 12.23.8022 and above
        - Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.23.8022 and above
        - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.23.8022 and above
        - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.23.8022 and above
        - Mellanox(R) ConnectX(R)-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.23.8022 and above
- ARM platforms with Mellanox(R) NICs combinations
    - CPU:
        - Qualcomm ARM 1.1 2500MHz
    - OS:
        - Red Hat Enterprise Linux Server release 7.5 (Maipo)
    - NICs:
        - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.24.0220
        - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.24.0220
- Mellanox(R) BlueField SmartNIC
    - Mellanox(R) BlueField SmartNIC MT416842 (2x25G)
        - Host interface: PCI Express 3.0 x16

        - Device ID: 15b3:a2d2
        - Firmware version: 18.24.0246
    - SoC ARM cores running OS:
        - CentOS Linux release 7.4.1708 (AltArch)
        - MLNX_OFED 4.4-2.5.3.0
    - DPDK application running on ARM cores inside SmartNIC
- ARM SoC combinations from NXP (with integrated NICs)
    - SoC:
        - NXP/Freescale QorIQ LS1046A with ARM Cortex A72
        - NXP/Freescale QorIQ LS2088A with ARM Cortex A72
    - OS:
        - Ubuntu 18.04.1 LTS with NXP QorIQ LSDK 1809 support packages
        - Ubuntu 16.04.3 LTS with NXP QorIQ LSDK 1803 support packages

CHAPTER

TWO

DPDK RELEASE 18.08

2.1 New Features

- **Added support for Hyper-V netvsc PMD.**

  The new `netvsc` poll mode driver provides native support for networking on Hyper-V. See the `../nics/netvsc` NIC driver guide for more details on this new driver.

- **Added Flow API support for CXGBE PMD.**

  Flow API support has been added to CXGBE Poll Mode Driver to offload flows to Chelsio T5/T6 NICs. Support added for:

  – Wildcard (LE-TCAM) and Exact (HASH) match filters.

  – Match items: physical ingress port, IPv4, IPv6, TCP and UDP.

  – Action items: queue, drop, count, and physical egress port redirect.

- **Added ixgbe preferred Rx/Tx parameters.**

  Rather than applications providing explicit Rx and Tx parameters such as queue and burst sizes, they can request that the EAL instead uses preferred values provided by the PMD, falling back to defaults within the EAL if the PMD does not provide any. The provision of such tuned values now includes the ixgbe PMD.

- **Added descriptor status check support for fm10k.**

  The `rte_eth_rx_descriptor_status` and `rte_eth_tx_descriptor_status` APIs are now supported by fm10K.

- **Updated the enic driver.**

  – Add low cycle count Tx handler for no-offload Tx.

  – Add low cycle count Rx handler for non-scattered Rx.

  – Minor performance improvements to scattered Rx handler.

  – Add handlers to add/delete VxLAN port number.

  – Add devarg to specify ingress VLAN rewrite mode.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  – Added port representors support.

  - Added Flow API support for e-switch rules. Added support for ACTION_PORT_ID, ACTION_DROP, ACTION_OF_POP_VLAN, ACTION_OF_PUSH_VLAN, ACTION_OF_SET_VLAN_VID, ACTION_OF_SET_VLAN_PCP and ITEM_PORT_ID.
  - Added support for 32-bit compilation.

- **Added TSO support for the mlx4 driver.**

  Added TSO support for the mlx4 drivers from MLNX_OFED_4.4 and above.

- **SoftNIC PMD rework.**

  The SoftNIC PMD infrastructure has been restructured to use the Packet Framework, which makes it more flexible, modular and easier to add new functionality in the future.

- **Updated the AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for:

  - 3DES for 8, 16 and 24 byte keys.

- **Added a new compression PMD using Intel's QuickAssist (QAT) device family.**

  Added the new `QAT` compression driver, for compression and decompression operations in software. See the `../compressdevs/qat_comp` compression driver guide for details on this new driver.

- **Updated the ISA-L PMD.**

  Added support for chained mbufs (input and output).

2.2 API Changes

- The path to the runtime config file has changed. The new path is determined as follows:
  - If DPDK is running as root, `/var/run/dpdk/<prefix>/config`
  - If DPDK is not running as root:
    - If `$XDG_RUNTIME_DIR` is set, `${XDG_RUNTIME_DIR}/dpdk/<prefix>/config`
    - Otherwise, `/tmp/dpdk/<prefix>/config`
- eal: The function `rte_eal_mbuf_default_mempool_ops` was deprecated and is removed in 18.08. It shall be replaced by `rte_mbuf_best_mempool_ops`.
- mempool: Following functions were deprecated and are removed in 18.08:
  - `rte_mempool_populate_iova_tab`
  - `rte_mempool_populate_phys_tab`
  - `rte_mempool_populate_phys` (`rte_mempool_populate_iova` should be used)
  - `rte_mempool_virt2phy` (`rte_mempool_virt2iova` should be used)
  - `rte_mempool_xmem_create`
  - `rte_mempool_xmem_size`
  - `rte_mempool_xmem_usage`

- ethdev: The old offload API is removed:
  - Rx per-port `rte_eth_conf.rxmode.[bit-fields]`
  - Tx per-queue `rte_eth_txconf.txq_flags`
  - `ETH_TXQ_FLAGS_NO*`

  The transition bits are removed:
  - `rte_eth_conf.rxmode.ignore_offload_bitfield`
  - `ETH_TXQ_FLAGS_IGNORE`
- cryptodev: The following API changes have been made in 18.08:
  - In struct `struct rte_cryptodev_info`, field `rte_pci_device *pci_dev` has been replaced with field `struct rte_device *device`.
  - Value 0 is accepted in `sym.max_nb_sessions`, meaning that a device supports an unlimited number of sessions.
  - Two new fields of type `uint16_t` have been added: `min_mbuf_headroom_req` and `min_mbuf_tailroom_req`. These parameters specify the recommended headroom and tailroom for mbufs to be processed by the PMD.
- cryptodev: The following functions were deprecated and are removed in 18.08:
  - `rte_cryptodev_queue_pair_start`
  - `rte_cryptodev_queue_pair_stop`
  - `rte_cryptodev_queue_pair_attach_sym_session`
  - `rte_cryptodev_queue_pair_detach_sym_session`
- cryptodev: The following functions were deprecated and are replaced by other functions in 18.08:
  - `rte_cryptodev_get_header_session_size` is replaced with `rte_cryptodev_sym_get_header_session_size`
  - `rte_cryptodev_get_private_session_size` is replaced with `rte_cryptodev_sym_get_private_session_size`
- cryptodev: Feature flag `RTE_CRYPTODEV_FF_MBUF_SCATTER_GATHER` is replaced with the following more explicit flags:
  - `RTE_CRYPTODEV_FF_IN_PLACE_SGL`
  - `RTE_CRYPTODEV_FF_OOP_SGL_IN_SGL_OUT`
  - `RTE_CRYPTODEV_FF_OOP_SGL_IN_LB_OUT`
  - `RTE_CRYPTODEV_FF_OOP_LB_IN_SGL_OUT`
  - `RTE_CRYPTODEV_FF_OOP_LB_IN_LB_OUT`
- cryptodev: Renamed cryptodev experimental APIs:

  Used user_data instead of private_data in following APIs to avoid confusion with the existing session parameter `sess_private_data[]` and related APIs.
  - `rte_cryptodev_sym_session_set_private_data()` changed to `rte_cryptodev_sym_session_set_user_data()`

  - `rte_cryptodev_sym_session_get_private_data()` changed to `rte_cryptodev_sym_session_get_user_data()`

- compressdev: Feature flag `RTE_COMP_FF_MBUF_SCATTER_GATHER` is replaced with the following more explicit flags:
  - `RTE_COMP_FF_OOP_SGL_IN_SGL_OUT`
  - `RTE_COMP_FF_OOP_SGL_IN_LB_OUT`
  - `RTE_COMP_FF_OOP_LB_IN_SGL_OUT`

2.3 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
   librte_bbdev.so.1
   librte_bitratestats.so.2
   librte_bpf.so.1
   librte_bus_dpaa.so.1
   librte_bus_fslmc.so.1
   librte_bus_pci.so.1
   librte_bus_vdev.so.1
+  librte_bus_vmbus.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
   librte_common_octeontx.so.1
   librte_compressdev.so.1
+  librte_cryptodev.so.5
   librte_distributor.so.1
+  librte_eal.so.8
+  librte_ethdev.so.10
+  librte_eventdev.so.5
   librte_flow_classify.so.1
   librte_gro.so.1
   librte_gso.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
   librte_latencystats.so.1
   librte_lpm.so.2
   librte_mbuf.so.4
+  librte_mempool.so.5
   librte_meter.so.2
   librte_metrics.so.1
   librte_net.so.1
   librte_pci.so.1
   librte_pdump.so.2
   librte_pipeline.so.3
   librte_pmd_bnxt.so.2
   librte_pmd_bond.so.2
   librte_pmd_i40e.so.2
   librte_pmd_ixgbe.so.2
   librte_pmd_dpaa2_cmdif.so.1
   librte_pmd_dpaa2_qdma.so.1
   librte_pmd_ring.so.2
   librte_pmd_softnic.so.1
   librte_pmd_vhost.so.2
```

```
librte_port.so.3
librte_power.so.1
librte_rawdev.so.1
librte_reorder.so.1
librte_ring.so.2
librte_sched.so.1
librte_security.so.1
librte_table.so.3
librte_timer.so.1
librte_vhost.so.3
```

2.4 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations
  - CPU
    - Intel(R) Atom(TM) CPU C3858 @ 2.00GHz
    - Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
    - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
    - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
    - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz
    - Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
    - Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz
  - OS:
    - CentOS 7.4
    - Fedora 25
    - Fedora 27
    - Fedora 28
    - FreeBSD 11.1
    - Red Hat Enterprise Linux Server release 7.5
    - SUSE Enterprise Linux 12
    - Wind River Linux 8
    - Ubuntu 14.04
    - Ubuntu 16.04
    - Ubuntu 16.10
    - Ubuntu 17.10
    - Ubuntu 18.04
  - NICs:

        * Intel(R) 82599ES 10 Gigabit Ethernet Controller
            - Firmware version: 0x61bf0001
            - Device id (pf/vf): 8086:10fb / 8086:10ed
            - Driver version: 5.2.3 (ixgbe)
        * Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
            - Firmware version: 0x800003e7
            - Device id (pf/vf): 8086:15ad / 8086:15a8
            - Driver version: 4.4.6 (ixgbe)
        * Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
            - Firmware version: 6.01 0x80003221
            - Device id (pf/vf): 8086:1572 / 8086:154c
            - Driver version: 2.4.6 (i40e)
        * Intel Corporation Ethernet Connection X722 for 10GbE SFP+ (4x10G)
            - Firmware version: 3.33 0x80000fd5 0.0.0
            - Device id (pf/vf): 8086:37d0 / 8086:37cd
            - Driver version: 2.4.3 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
            - Firmware version: 6.01 0x80003221
            - Device id (pf/vf): 8086:158b / 8086:154c
            - Driver version: 2.4.6 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
            - Firmware version: 6.01 0x8000321c
            - Device id (pf/vf): 8086:1583 / 8086:154c
            - Driver version: 2.4.6 (i40e)
        * Intel(R) Corporation I350 Gigabit Network Connection
            - Firmware version: 1.63, 0x80000dda
            - Device id (pf/vf): 8086:1521 / 8086:1520
            - Driver version: 5.4.0-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
    - CPU:
        * Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
        * Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz

    - Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
    - Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
    - Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
  - OS:
    - Red Hat Enterprise Linux Server release 7.5 (Maipo)
    - Red Hat Enterprise Linux Server release 7.4 (Maipo)
    - Red Hat Enterprise Linux Server release 7.3 (Maipo)
    - Red Hat Enterprise Linux Server release 7.2 (Maipo)
    - Ubuntu 18.04
    - Ubuntu 17.10
    - Ubuntu 16.04
    - SUSE Linux Enterprise Server 15
  - MLNX_OFED: 4.3-2.0.2.0
  - MLNX_OFED: 4.4-2.0.1.0
  - NICs:
    - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1007
      - Firmware version: 2.42.5000
    - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013

      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000 and above

        * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.21.1000 and above
        * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.21.1000 and above
        * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.21.1000 and above
- ARM platforms with Mellanox(R) NICs combinations
    - CPU:
        * Qualcomm ARM 1.1 2500MHz
    - OS:
        * Red Hat Enterprise Linux Server release 7.5 (Maipo)
    - NICs:
        * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.23.1000
        * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.23.1000
- Mellanox BlueField SmartNIC
    - **Mellanox(R) BlueField SmartNIC MT416842 (2x25G)**
        * Host interface: PCI Express 3.0 x16
        * Device ID: 15b3:a2d2
        * Firmware version: 18.99.3950
    - SoC ARM cores running OS: * CentOS Linux release 7.4.1708 (AltArch) * Mellanox MLNX_OFED 4.2-1.4.21.0
    - DPDK application running on ARM cores inside SmartNIC
    - Bluefield representors support planned for next release.

CHAPTER

THREE

DPDK RELEASE 18.05

3.1 New Features

- **Reworked memory subsystem.**

  Memory subsystem has been reworked to support new functionality.

  On Linux, support for reserving/unreserving hugepage memory at runtime has been added, so applications no longer need to pre-reserve memory at startup. Due to reorganized internal workings of memory subsystem, any memory allocated through `rte_malloc()` or `rte_memzone_reserve()` is no longer guaranteed to be IOVA-contiguous.

  This functionality has introduced the following changes:

  - `rte_eal_get_physmem_layout()` was removed.
  - A new flag for memzone reservation (`RTE_MEMZONE_IOVA_CONTIG`) was added to ensure reserved memory will be IOVA-contiguous, for use with device drivers and other cases requiring such memory.
  - New callbacks for memory allocation/deallocation events, allowing users (or drivers) to be notified of new memory being allocated or deallocated
  - New callbacks for validating memory allocations above a specified limit, allowing user to permit or deny memory allocations.
  - A new command-line switch `--legacy-mem` to enable EAL behavior similar to how older versions of DPDK worked (memory segments that are IOVA-contiguous, but hugepages are reserved at startup only, and can never be released).
  - A new command-line switch `--single-file-segments` to put all memory segments within a segment list in a single file.
  - A set of convenience function calls to look up and iterate over allocated memory segments.
  - `-m` and `--socket-mem` command-line arguments now carry an additional meaning and mark pre-reserved hugepages as "unfree-able", thereby acting as a mechanism guaranteeing minimum availability of hugepage memory to the application.

  Reserving/unreserving memory at runtime is not currently supported on FreeBSD.

- **Added bucket mempool driver.**

  Added a bucket mempool driver which provides a way to allocate contiguous block of objects. The number of objects in the block depends on how many objects fit in the

RTE_DRIVER_MEMPOOL_BUCKET_SIZE_KB memory chunk which is a build time option. The number may be obtained using rte_mempool_ops_get_info() API. Contiguous blocks may be allocated using rte_mempool_get_contig_blocks() API.

- **Added support for port representors.**

  Added DPDK port representors (also known as "VF representors" in the specific context of VFs), which are to DPDK what the Ethernet switch device driver model (**switchdev**) is to Linux, and which can be thought as a software "patch panel" front-end for applications. DPDK port representors are implemented as additional virtual Ethernet device (**ethdev**) instances, spawned on an as-needed basis through configuration parameters passed to the driver of the underlying device using devargs.

- **Added support for VXLAN and NVGRE tunnel endpoint.**

  New actions types have been added to support encapsulation and decapsulation operations for a tunnel endpoint. The new action types are RTE_FLOW_ACTION_TYPE_[VXLAN/NVGRE]_ENCAP, RTE_FLOW_ACTION_TYPE_[VXLAN/NVGRE]_DECAP, RTE_FLOW_ACTION_TYPE_JUMP. A new item type RTE_FLOW_ACTION_TYPE_MARK has been added to match a flow against a previously marked flow. A shared counter has also been introduced to the flow API to count a group of flows.

- **Added PMD-recommended Tx and Rx parameters.**

  Applications can now query drivers for device-tuned values of ring sizes, burst sizes, and number of queues.

- **Added RSS hash and key update to CXGBE PMD.**

  Added support for updating the RSS hash and key to the CXGBE PMD.

- **Added CXGBE VF PMD.**

  CXGBE VF Poll Mode Driver has been added to run DPDK over Chelsio T5/T6 NIC VF instances.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  – Introduced Multi-packet Rx to enable 100Gb/sec with 64B frames.

  – Support for being run by non-root users given a reduced set of capabilities CAP_NET_ADMIN, CAP_NET_RAW and CAP_IPC_LOCK.

  – Support for TSO and checksum for generic UDP and IP tunnels.

  – Support for inner checksum and RSS for GRE, VXLAN-GPE, MPLSoGRE and MPLSoUDP tunnels.

  – Accommodate the new memory hotplug model.

  – Support for non virtually contiguous mempools.

  – Support for MAC adding along with allmulti and promiscuous modes from VF.

  – Support for Mellanox BlueField SoC device.

  – Support for PMD defaults for queue number and depth to improve the out of the box performance.

- **Updated mlx4 driver.**

  Updated the mlx4 driver including the following changes:

  - Support for to being run by non-root users given a reduced set of capabilities `CAP_NET_ADMIN`, `CAP_NET_RAW` and `CAP_IPC_LOCK`.
  - Supported CRC strip toggling.
  - Accommodate the new memory hotplug model.
  - Support non virtually contiguous mempools.
  - Dropped support for Mellanox OFED 4.2.

- **Updated Solarflare network PMD.**

  Updated the sfc_efx driver including the following changes:

  - Added support for Solarflare XtremeScale X2xxx family adapters.
  - Added support for NVGRE, VXLAN and GENEVE filters in flow API.
  - Added support for DROP action in flow API.
  - Added support for equal stride super-buffer Rx mode (X2xxx only).
  - Added support for MARK and FLAG actions in flow API (X2xxx only).

- **Added Ethernet poll mode driver for AMD XGBE devices.**

  Added the new `axgbe` ethernet poll mode driver for AMD XGBE devices. See the `../nics/axgbe` nic driver guide for more details on this new driver.

- **Updated szedata2 PMD.**

  Added support for new NFB-200G2QL card. A new API was introduced in the libsze2 library which the szedata2 PMD depends on, thus the new version of the library was needed. New versions of the packages are available and the minimum required version is 4.4.1.

- **Added support for Broadcom NetXtreme-S (BCM58800) family of controllers (aka Stingray).**

  Added support for the Broadcom NetXtreme-S (BCM58800) family of controllers (aka Stingray). The BCM58800 devices feature a NetXtreme E-Series advanced network controller, a high-performance ARM CPU block, PCI Express (PCIe) Gen3 interfaces, key accelerators for compute offload and a high-speed memory subsystem including L3 cache and DDR4 interfaces, all interconnected by a coherent Network-on-chip (NOC) fabric.

  The ARM CPU subsystem features eight ARMv8 Cortex-A72 CPUs at 3.0 GHz, arranged in a multi-cluster configuration.

- **Added vDPA in vhost-user lib.**

  Added support for selective datapath in the vhost-user lib. vDPA stands for vhost Data Path Acceleration. It supports virtio ring compatible devices to serve the virtio driver directly to enable datapath acceleration.

- **Added IFCVF vDPA driver.**

Added IFCVF vDPA driver to support Intel FPGA 100G VF devices. IFCVF works as a HW vhost data path accelerator, it supports live migration and is compatible with virtio 0.95 and 1.0. This driver registers the ifcvf vDPA driver to vhost lib, when virtio connects. With the help of the registered vDPA driver the assigned VF gets configured to Rx/Tx directly to VM's virtio vrings.

- **Added support for vhost dequeue interrupt mode.**

  Added support for vhost dequeue interrupt mode to release CPUs to others when there is no data to transmit. Applications can register an epoll event file descriptor to associate Rx queues with interrupt vectors.

- **Added support for virtio-user server mode.**

  In a container environment if the vhost-user backend restarts, there's no way for it to reconnect to virtio-user. To address this, support for server mode has been added. In this mode the socket file is created by virtio-user, which the backend connects to. This means that if the backend restarts, it can reconnect to virtio-user and continue communications.

- **Added crypto workload support to vhost library.**

  New APIs have been introduced in the vhost library to enable virtio crypto support including session creation/deletion handling and translating virtio-crypto requests into DPDK crypto operations. A sample application has also been introduced.

- **Added virtio crypto PMD.**

  Added a new Poll Mode Driver for virtio crypto devices, which provides AES-CBC ciphering and AES-CBC with HMAC-SHA1 algorithm-chaining. See the `../cryptodevs/virtio` crypto driver guide for more details on this new driver.

- **Added AMD CCP Crypto PMD.**

  Added the new `ccp` crypto driver for AMD CCP devices. See the `../cryptodevs/ccp` crypto driver guide for more details on this new driver.

- **Updated AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for:

  - AES-CMAC (128-bit key).

- **Added the Compressdev Library, a generic compression service library.**

  Added the Compressdev library which provides an API for offload of compression and decompression operations to hardware or software accelerator devices.

- **Added a new compression poll mode driver using Intels ISA-L.**

  Added the new `ISA-L` compression driver, for compression and decompression operations in software. See the `../compressdevs/isal` compression driver guide for details on this new driver.

- **Added the Event Timer Adapter Library.**

  The Event Timer Adapter Library extends the event-based model by introducing APIs that allow applications to arm/cancel event timers that generate timer expiry events. This new type of event is scheduled by an event device along with existing types of events.

- **Added OcteonTx TIM Driver (Event timer adapter).**

The OcteonTx Timer block enables software to schedule events for a future time, it is exposed to an application via the Event timer adapter library.

See the `../eventdevs/octeontx` guide for more details

- **Added Event Crypto Adapter Library.**

  Added the Event Crypto Adapter Library. This library extends the event-based model by introducing APIs that allow applications to enqueue/dequeue crypto operations to/from cryptodev as events scheduled by an event device.

- **Added Ifpga Bus, a generic Intel FPGA Bus library.**

  Added the Ifpga Bus library which provides support for integrating any Intel FPGA device with the DPDK framework. It provides Intel FPGA Partial Bit Stream AFU (Accelerated Function Unit) scan and drivers probe.

- **Added IFPGA (Intel FPGA) Rawdev Driver.**

  Added a new Rawdev driver called IFPGA (Intel FPGA) Rawdev Driver, which cooperates with OPAE (Open Programmable Acceleration Engine) shared code to provide common FPGA management ops for FPGA operation.

  See the `../rawdevs/ifpga_rawdev` programmer's guide for more details.

- **Added DPAA2 QDMA Driver (in rawdev).**

  The DPAA2 QDMA is an implementation of the rawdev API, that provide a means of initiating a DMA transaction from CPU. The initiated DMA is performed without the CPU being involved in the actual DMA transaction.

  See the `../rawdevs/dpaa2_qdma` guide for more details.

- **Added DPAA2 Command Interface Driver (in rawdev).**

  The DPAA2 CMDIF is an implementation of the rawdev API, that provides communication between the GPP and NXP's QorIQ based AIOP Block (Firmware). Advanced IO Processor i.e. AIOP are clusters of programmable RISC engines optimized for flexible networking and I/O operations. The communication between GPP and AIOP is achieved via using DPCI devices exposed by MC for GPP <–> AIOP interaction.

  See the `../rawdevs/dpaa2_cmdif` guide for more details.

- **Added device event monitor framework.**

  Added a general device event monitor framework to EAL, for device dynamic management to facilitate device hotplug awareness and associated actions. The list of new APIs is:

  - `rte_dev_event_monitor_start` and `rte_dev_event_monitor_stop` for the event monitor enabling and disabling.
  - `rte_dev_event_callback_register` and `rte_dev_event_callback_unregister` for registering and un-registering user callbacks.

  Linux uevent is supported as a backend of this device event notification framework.

- **Added support for procinfo and pdump on eth vdev.**

  For ethernet virtual devices (like TAP, PCAP, etc.), with this feature, we can get stats/xstats on shared memory from a secondary process, and also pdump packets on those virtual devices.

- **Enhancements to the Packet Framework Library.**

  Design and development of new API functions for Packet Framework library that implement a common set of actions such as traffic metering, packet encapsulation, network address translation, TTL update, etc., for pipeline table and input ports to speed up application development. The API functions includes creating action profiles, registering actions to the profiles, instantiating action profiles for pipeline table and input ports, etc.

- **Added the BPF Library.**

  The BPF Library provides the ability to load and execute Enhanced Berkeley Packet Filters (eBPF) within user-space DPDK applications. It also introduces a basic framework to load/unload BPF-based filters on Eth devices (right now only via SW RX/TX callbacks). It also adds a dependency on libelf.

3.2 API Changes

- service cores: No longer marked as experimental.

  The service cores functions are no longer marked as experimental, and have become part of the normal DPDK API and ABI. Any future ABI changes will be announced at least one release before the ABI change is made. There are no ABI breaking changes planned.

- eal: The `rte_lcore_has_role()` return value changed.

  This function now returns true or false, respectively, rather than 0 or < 0 for success or failure. It makes use of the function more intuitive.

- mempool: The capability flags and related functions have been removed.

  Flags `MEMPOOL_F_CAPA_PHYS_CONTIG` and `MEMPOOL_F_CAPA_BLK_ALIGNED_OBJECTS` were used by octeontx mempool driver to customize generic mempool library behavior. Now the new driver callbacks `calc_mem_size` and `populate` may be used to achieve it without specific knowledge in the generic code.

- mempool: The following xmem functions have been deprecated:

  - `rte_mempool_xmem_create`
  - `rte_mempool_xmem_size`
  - `rte_mempool_xmem_usage`
  - `rte_mempool_populate_iova_tab`

- mbuf: The control mbuf API has been removed in v18.05. The impacted functions and macros are:

  - `rte_ctrlmbuf_init()`
  - `rte_ctrlmbuf_alloc()`
  - `rte_ctrlmbuf_free()`
  - `rte_ctrlmbuf_data()`
  - `rte_ctrlmbuf_len()`
  - `rte_is_ctrlmbuf()`

  - CTRL_MBUF_FLAG

  The packet mbuf API should be used as a replacement.

- meter: API updated to accommodate configuration profiles.

  The meter API has been changed to support meter configuration profiles. The configuration profile represents the set of configuration parameters for a given meter object, such as the rates and sizes for the token buckets. These configuration parameters were previously part of the meter object internal data structure. The separation of the configuration parameters from the meter object data structure results in reducing its memory footprint which helps in better cache utilization when a large number of meter objects are used.

- ethdev: The function `rte_eth_dev_count()`, often mis-used to iterate over ports, is deprecated and replaced by `rte_eth_dev_count_avail()`. There is also a new function `rte_eth_dev_count_total()` to get the total number of allocated ports, available or not. The hotplug-proof applications should use `RTE_ETH_FOREACH_DEV` or `RTE_ETH_FOREACH_DEV_OWNED_BY` as port iterators.

- ethdev: In struct `struct rte_eth_dev_info`, field `rte_pci_device *pci_dev` has been replaced with field `struct rte_device *device`.

- ethdev: Changes to the semantics of `rte_eth_dev_configure()` parameters.

  If both the `nb_rx_q` and `nb_tx_q` parameters are zero, `rte_eth_dev_configure()` will now use PMD-recommended queue sizes, or if recommendations are not provided by the PMD the function will use ethdev fall-back values. Previously setting both of the parameters to zero would have resulted in `-EINVAL` being returned.

- ethdev: Changes to the semantics of `rte_eth_rx_queue_setup()` parameters.

  If the `nb_rx_desc` parameter is zero, `rte_eth_rx_queue_setup` will now use the PMD-recommended Rx ring size, or in the case where the PMD does not provide a recommendation, will use an ethdev-provided fall-back value. Previously, setting `nb_rx_desc` to zero would have resulted in an error.

- ethdev: Changes to the semantics of `rte_eth_tx_queue_setup()` parameters.

  If the `nb_tx_desc` parameter is zero, `rte_eth_tx_queue_setup` will now use the PMD-recommended Tx ring size, or in the case where the PMD does not provide a recommendation, will use an ethdev-provided fall-back value. Previously, setting `nb_tx_desc` to zero would have resulted in an error.

- ethdev: Several changes were made to the flow API.

  - The unused DUP action was removed.
  - Actions semantics in flow rules: list order now matters ("first to last" instead of "all simultaneously"), repeated actions are now all performed, and they do not individually have (non-)terminating properties anymore.
  - Flow rules are now always terminating unless a `PASSTHRU` action is present.
  - C99-style flexible arrays were replaced with standard pointers in RSS action and in RAW pattern item structures due to compatibility issues.
  - The RSS action was modified to not rely on external `struct rte_eth_rss_conf` anymore to instead expose its own and more appropriately named configuration fields directly (`rss_conf->rss_key` => `key`, `rss_conf->rss_key_len` => `key_len`, `rss_conf->rss_hf` => `types`, `num` => `queue_num`), and the addition

of missing RSS parameters (`func` for RSS hash function to apply and `level` for the encapsulation level).

  - The VLAN pattern item (`struct rte_flow_item_vlan`) was modified to include inner EtherType instead of outer TPID. Its default mask was also modified to cover the VID part (lower 12 bits) of TCI only.
  - A new transfer attribute was added to `struct rte_flow_attr` in order to clarify the behavior of some pattern items.
  - PF and VF pattern items are now only accepted by PMDs that implement them (bnxt and i40e) when the transfer attribute is also present, for consistency.
  - Pattern item PORT was renamed PHY_PORT to avoid confusion with DPDK port IDs.
  - An action counterpart to the PHY_PORT pattern item was added in order to redirect matching traffic to a specific physical port.
  - PORT_ID pattern item and actions were added to match and target DPDK port IDs at a higher level than PHY_PORT.
  - `RTE_FLOW_ACTION_TYPE_[VXLAN/NVGRE]_ENCAP` action items were added to support tunnel encapsulation operation for VXLAN and NVGRE type tunnel endpoint.
  - `RTE_FLOW_ACTION_TYPE_[VXLAN/NVGRE]_DECAP` action items were added to support tunnel decapsulation operation for VXLAN and NVGRE type tunnel endpoint.
  - `RTE_FLOW_ACTION_TYPE_JUMP` action item was added to support a matched flow to be redirected to the specific group.
  - `RTE_FLOW_ACTION_TYPE_MARK` item type has been added to match a flow against a previously marked flow.

- ethdev: Change flow APIs regarding count action:
  - `rte_flow_create()` API count action now requires the `struct rte_flow_action_count`.
  - `rte_flow_query()` API parameter changed from action type to action structure.
- ethdev: Changes to offload API

  A pure per-port offloading isn't requested to be repeated in [rt]x_conf->offloads to `rte_eth_[rt]x_queue_setup()`. Now any offloading enabled in `rte_eth_dev_configure()` can't be disabled by `rte_eth_[rt]x_queue_setup()`. Any new added offloading which has not been enabled in `rte_eth_dev_configure()` and is requested to be enabled in `rte_eth_[rt]x_queue_setup()` must be per-queue type, or otherwise trigger an error log.

- ethdev: Runtime queue setup

  `rte_eth_rx_queue_setup` and `rte_eth_tx_queue_setup` can be called after `rte_eth_dev_start` if the device supports runtime queue setup. The device driver can expose this capability through `rte_eth_dev_info_get`. A Rx or Tx queue set up at runtime need to be started explicitly by `rte_eth_dev_rx_queue_start` or `rte_eth_dev_tx_queue_start`.

3.3 ABI Changes

- ring: The alignment constraints on the ring structure has been relaxed to one cache line instead of two, and an empty cache line padding is added between the producer and consumer structures. The size of the structure and the offset of the fields remains the same on platforms with 64B cache line, but changes on other platforms.
- mempool: Some ops have changed.

  A new callback `calc_mem_size` has been added to `rte_mempool_ops` to allow customization of the required memory size calculation. A new callback `populate` has been added to `rte_mempool_ops` to allow customized object population. Callback `get_capabilities` has been removed from `rte_mempool_ops` since its features are covered by `calc_mem_size` and `populate` callbacks. Callback `register_memory_area` has been removed from `rte_mempool_ops` since the new callback `populate` may be used instead of it.
- ethdev: Additional fields in rte_eth_dev_info.

  The `rte_eth_dev_info` structure has had two extra entries appended to the end of it: `default_rxportconf` and `default_txportconf`. Each of these in turn are `rte_eth_dev_portconf` structures containing three fields of type `uint16_t`: `burst_size`, `ring_size`, and `nb_queues`. These are parameter values recommended for use by the PMD.
- ethdev: ABI for all flow API functions was updated.

  This includes functions `rte_flow_copy`, `rte_flow_create`, `rte_flow_destroy`, `rte_flow_error_set`, `rte_flow_flush`, `rte_flow_isolate`, `rte_flow_query` and `rte_flow_validate`, due to changes in error type definitions (`enum rte_flow_error_type`), removal of the unused DUP action (`enum rte_flow_action_type`), modified behavior for flow rule actions (see API changes), removal of C99 flexible array from RAW pattern item (`struct rte_flow_item_raw`), complete rework of the RSS action definition (`struct rte_flow_action_rss`), sanity fix in the VLAN pattern item (`struct rte_flow_item_vlan`) and new transfer attribute (`struct rte_flow_attr`).
- bbdev: New parameter added to rte_bbdev_op_cap_turbo_dec.

  A new parameter `max_llr_modulus` has been added to `rte_bbdev_op_cap_turbo_dec` structure to specify maximal LLR (likelihood ratio) absolute value.
- bbdev: Queue Groups split into UL/DL Groups.

  Queue Groups have been split into UL/DL Groups in the Turbo Software Driver. They are independent for Decode/Encode. `rte_bbdev_driver_info` reflects introduced changes.

3.4 Known Issues

- **Secondary process launch is not reliable.**

  Recent memory hotplug patches have made multiprocess startup less reliable than it was in past releases. A number of workarounds are known to work depending on the

circumstances. As such it isn't recommended to use the secondary process mechanism for critical systems. The underlying issues will be addressed in upcoming releases.

The issue is explained in more detail, including potential workarounds, in the Bugzilla entry referenced below.

Bugzilla entry: https://bugs.dpdk.org/show_bug.cgi?id=50

- **pdump is not compatible with old applications.**

  As we changed to use generic multi-process communication for pdump negotiation instead of previous dedicated unix socket way, pdump applications, including the dpdk-pdump example and any other applications using `librte_pdump`, will not work with older version DPDK primary applications.

- **rte_abort takes a long time on FreeBSD.**

  DPDK processes now allocates a large area of virtual memory address space. As a result `rte_abort` on FreeBSD now dumps the contents of the whole reserved memory range, not just the used portion, to a core dump file. Writing this large core file can take a significant amount of time, causing processes to appear to hang on the system.

  The work around for the issue is to set the system resource limits for core dumps before running any tests, e.g. `limit coredumpsize 0`. This will effectively disable core dumps on FreeBSD. If they are not to be completely disabled, a suitable limit, e.g. 1G might be specified instead of 0. This needs to be run per-shell session, or before every test run. This change can also be made persistent by adding `kern.coredump=0` to `/etc/sysctl.conf`.

  Bugzilla entry: https://bugs.dpdk.org/show_bug.cgi?id=53

- **ixgbe PMD crash on hotplug detach when no VF created.**

  ixgbe PMD uninit path cause null pointer dereference because of port representor cleanup when number of VF is zero.

  Bugzilla entry: https://bugs.dpdk.org/show_bug.cgi?id=57

- **Bonding PMD may fail to accept new slave ports in certain conditions.**

  In certain conditions when using testpmd, bonding may fail to register new slave ports.

  Bugzilla entry: https://bugs.dpdk.org/show_bug.cgi?id=52.

- **Unexpected performance regression in Vhost library.**

  Patches fixing CVE-2018-1059 were expected to introduce a small performance drop. However, in some setups, bigger performance drops have been measured when running micro-benchmarks.

  Bugzilla entry: https://bugs.dpdk.org/show_bug.cgi?id=48

3.5 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
   librte_bbdev.so.1
   librte_bitratestats.so.2
+  librte_bpf.so.1
```

```
   librte_bus_dpaa.so.1
   librte_bus_fslmc.so.1
   librte_bus_pci.so.1
   librte_bus_vdev.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
+  librte_common_octeontx.so.1
+  librte_compressdev.so.1
   librte_cryptodev.so.4
   librte_distributor.so.1
+  librte_eal.so.7
+  librte_ethdev.so.9
+  librte_eventdev.so.4
   librte_flow_classify.so.1
   librte_gro.so.1
   librte_gso.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
   librte_latencystats.so.1
   librte_lpm.so.2
+  librte_mbuf.so.4
+  librte_mempool.so.4
+  librte_meter.so.2
   librte_metrics.so.1
   librte_net.so.1
   librte_pci.so.1
   librte_pdump.so.2
   librte_pipeline.so.3
   librte_pmd_bnxt.so.2
   librte_pmd_bond.so.2
   librte_pmd_i40e.so.2
   librte_pmd_ixgbe.so.2
+  librte_pmd_dpaa2_cmdif.so.1
+  librte_pmd_dpaa2_qdma.so.1
   librte_pmd_ring.so.2
   librte_pmd_softnic.so.1
   librte_pmd_vhost.so.2
   librte_port.so.3
   librte_power.so.1
   librte_rawdev.so.1
   librte_reorder.so.1
+  librte_ring.so.2
   librte_sched.so.1
   librte_security.so.1
   librte_table.so.3
   librte_timer.so.1
   librte_vhost.so.3
```

3.6 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations
  - CPU
    - Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
    - Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz

    - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
    - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
    - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz
    - Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
    - Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz
  - OS:
    - CentOS 7.4
    - Fedora 25
    - Fedora 27
    - Fedora 28
    - FreeBSD 11.1
    - Red Hat Enterprise Linux Server release 7.3
    - SUSE Enterprise Linux 12
    - Wind River Linux 8
    - Ubuntu 14.04
    - Ubuntu 16.04
    - Ubuntu 16.10
    - Ubuntu 17.10
  - NICs:
    - Intel(R) 82599ES 10 Gigabit Ethernet Controller
      - Firmware version: 0x61bf0001
      - Device id (pf/vf): 8086:10fb / 8086:10ed
      - Driver version: 5.2.3 (ixgbe)
    - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
      - Firmware version: 0x800003e7
      - Device id (pf/vf): 8086:15ad / 8086:15a8
      - Driver version: 4.4.6 (ixgbe)
    - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
      - Firmware version: 6.01 0x80003221
      - Device id (pf/vf): 8086:1572 / 8086:154c
      - Driver version: 2.4.6 (i40e)
    - Intel Corporation Ethernet Connection X722 for 10GbE SFP+ (4x10G)

            - Firmware version: 3.33 0x80000fd5 0.0.0
            - Device id (pf/vf): 8086:37d0 / 8086:37cd
            - Driver version: 2.4.3 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
            - Firmware version: 6.01 0x80003221
            - Device id (pf/vf): 8086:158b / 8086:154c
            - Driver version: 2.4.6 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
            - Firmware version: 6.01 0x8000321c
            - Device id (pf/vf): 8086:1583 / 8086:154c
            - Driver version: 2.4.6 (i40e)
        * Intel(R) Corporation I350 Gigabit Network Connection
            - Firmware version: 1.63, 0x80000dda
            - Device id (pf/vf): 8086:1521 / 8086:1520
            - Driver version: 5.4.0-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
    - CPU:
        * Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
        * Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
        * Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
        * Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
        * Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
    - OS:
        * Red Hat Enterprise Linux Server release 7.5 (Maipo)
        * Red Hat Enterprise Linux Server release 7.4 (Maipo)
        * Red Hat Enterprise Linux Server release 7.3 (Maipo)
        * Red Hat Enterprise Linux Server release 7.2 (Maipo)
        * Ubuntu 18.04
        * Ubuntu 17.10
        * Ubuntu 16.10
        * Ubuntu 16.04
        * SUSE Linux Enterprise Server 15

- MLNX_OFED: 4.2-1.0.0.0
- MLNX_OFED: 4.3-2.0.2.0
- NICs:
  - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1007
    - Firmware version: 2.42.5000
  - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.21.1000 and above
  - Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013

      - Firmware version: 12.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
      - Firmware version: 12.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000 and above
  * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000 and above
  * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1017
      - Firmware version: 16.21.1000 and above
  * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
      - Host interface: PCI Express 4.0 x16
      - Device ID: 15b3:1019
      - Firmware version: 16.21.1000 and above

- ARM platforms with Mellanox(R) NICs combinations
  - CPU:
    - Qualcomm ARM 1.1 2500MHz
  - OS:
    - Red Hat Enterprise Linux Server release 7.5 (Maipo)
  - NICs:
    - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.22.0428
    - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1017
      - Firmware version: 16.22.0428
- ARM SoC combinations from Cavium (with integrated NICs)
  - SoC:
    - Cavium CN81xx
    - Cavium CN83xx
  - OS:
    - Ubuntu 16.04.2 LTS with Cavium SDK-6.2.0-Patch2 release support package.
- ARM SoC combinations from NXP (with integrated NICs)
  - SoC:
    - NXP/Freescale QorIQ LS1046A with ARM Cortex A72
    - NXP/Freescale QorIQ LS2088A with ARM Cortex A72
  - OS:
    - Ubuntu 16.04.3 LTS with NXP QorIQ LSDK 1803 support packages

CHAPTER

FOUR

DPDK RELEASE 18.02

4.1 New Features

- **Added function to allow releasing internal EAL resources on exit.**

  During `rte_eal_init()` EAL allocates memory from hugepages to enable its core libraries to perform their tasks. The `rte_eal_cleanup()` function releases these resources, ensuring that no hugepage memory is leaked. It is expected that all DPDK applications call `rte_eal_cleanup()` before exiting. Not calling this function could result in leaking hugepages, leading to failure during initialization of secondary processes.

- **Added igb, ixgbe and i40e ethernet driver to support RSS with flow API.**

  Added support for igb, ixgbe and i40e NICs with existing RSS configuration using the `rte_flow` API.

  Also enabled queue region configuration using the `rte_flow` API for i40e.

- **Updated i40e driver to support PPPoE/PPPoL2TP.**

  Updated i40e PMD to support PPPoE/PPPoL2TP with PPPoE/PPPoL2TP supporting profiles which can be programmed by dynamic device personalization (DDP) process.

- **Added MAC loopback support for i40e.**

  Added MAC loopback support for i40e in order to support test tasks requested by users. It will setup `Tx -> Rx` loopback link according to the device configuration.

- **Added support of run time determination of number of queues per i40e VF.**

  The number of queue per VF is determined by its host PF. If the PCI address of an i40e PF is `aaaa:bb.cc`, the number of queues per VF can be configured with EAL parameter like `-w aaaa:bb.cc,queue-num-per-vf=n`. The value n can be 1, 2, 4, 8 or 16. If no such parameter is configured, the number of queues per VF is 4 by default.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  – Enabled compilation as a plugin, thus removed the mandatory dependency with rdma-core. With the special compilation, the rdma-core libraries will be loaded only in case Mellanox device is being used. For binaries creation the PMD can be enabled, still not requiring from every end user to install rdma-core.

  – Improved multi-segment packet performance.

  - Changed driver name to use the PCI address to be compatible with OVS-DPDK APIs.
  - Extended statistics for physical port packet/byte counters.
  - Converted to the new offloads API.
  - Supported device removal check operation.

- **Updated mlx4 driver.**

  Updated the mlx4 driver including the following changes:

  - Enabled compilation as a plugin, thus removed the mandatory dependency with rdma-core. With the special compilation, the rdma-core libraries will be loaded only in case Mellanox device is being used. For binaries creation the PMD can be enabled, still not requiring from every end user to install rdma-core.
  - Improved data path performance.
  - Converted to the new offloads API.
  - Supported device removal check operation.

- **Added NVGRE and UDP tunnels support in Solarflare network PMD.**

  Added support for NVGRE, VXLAN and GENEVE tunnels.

  - Added support for UDP tunnel ports configuration.
  - Added tunneled packets classification.
  - Added inner checksum offload.

- **Added AVF (Adaptive Virtual Function) net PMD.**

  Added a new net PMD called AVF (Adaptive Virtual Function), which supports Intel® Ethernet Adaptive Virtual Function (AVF) with features such as:

  - Basic Rx/Tx burst
  - SSE vectorized Rx/Tx burst
  - Promiscuous mode
  - MAC/VLAN offload
  - Checksum offload
  - TSO offload
  - Jumbo frame and MTU setting
  - RSS configuration
  - stats
  - Rx/Tx descriptor status
  - Link status update/event

- **Added feature supports for live migration from vhost-net to vhost-user.**

  Added feature supports for vhost-user to make live migration from vhost-net to vhost-user possible. The features include:

  - VIRTIO_F_ANY_LAYOUT
  - VIRTIO_F_EVENT_IDX
  - VIRTIO_NET_F_GUEST_ECN, VIRTIO_NET_F_HOST_ECN
  - VIRTIO_NET_F_GUEST_UFO, VIRTIO_NET_F_HOST_UFO
  - VIRTIO_NET_F_GSO

  Also added VIRTIO_NET_F_GUEST_ANNOUNCE feature support in virtio pmd. In a scenario where the vhost backend doesn't have the ability to generate RARP packets, the VM running virtio pmd can still be live migrated if VIRTIO_NET_F_GUEST_ANNOUNCE feature is negotiated.

- **Updated the AESNI-MB PMD.**

  The AESNI-MB PMD has been updated with additional support for:

  - AES-CCM algorithm.

- **Updated the DPAA_SEC crypto driver to support rte_security.**

  Updated the dpaa_sec crypto PMD to support rte_security lookaside protocol offload for IPsec.

- **Added Wireless Base Band Device (bbdev) abstraction.**

  The Wireless Baseband Device library is an acceleration abstraction framework for 3gpp Layer 1 processing functions that provides a common programming interface for seamless operation on integrated or discrete hardware accelerators or using optimized software libraries for signal processing.

  The current release only supports 3GPP CRC, Turbo Coding and Rate Matching operations, as specified in 3GPP TS 36.212.

  See the ../prog_guide/bbdev programmer's guide for more details.

- **Added New eventdev Ordered Packet Distribution Library (OPDL) PMD.**

  The OPDL (Ordered Packet Distribution Library) eventdev is a specific implementation of the eventdev API. It is particularly suited to packet processing workloads that have high throughput and low latency requirements. All packets follow the same path through the device. The order in which packets follow is determined by the order in which queues are set up. Events are left on the ring until they are transmitted. As a result packets do not go out of order.

  With this change, applications can use the OPDL PMD via the eventdev api.

- **Added new pipeline use case for dpdk-test-eventdev application.**

  Added a new "pipeline" use case for the dpdk-test-eventdev application. The pipeline case can be used to simulate various stages in a real world application from packet receive to transmit while maintaining the packet ordering. It can also be used to measure the performance of the event device across the stages of the pipeline.

  The pipeline use case has been made generic to work with all the event devices based on the capabilities.

- **Updated Eventdev sample application to support event devices based on capability.**

Updated the Eventdev pipeline sample application to support various types of pipelines based on the capabilities of the attached event and ethernet devices. Also, renamed the application from software PMD specific `eventdev_pipeline_sw_pmd` to the more generic `eventdev_pipeline`.

- **Added Rawdev, a generic device support library.**

  The Rawdev library provides support for integrating any generic device type with the DPDK framework. Generic devices are those which do not have a pre-defined type within DPDK, for example, ethernet, crypto, event etc.

  A set of northbound APIs have been defined which encompass a generic set of operations by allowing applications to interact with device using opaque structures/buffers. Also, southbound APIs provide a means of integrating devices either as as part of a physical bus (PCI, FSLMC etc) or through `vdev`.

  See the `../prog_guide/rawdev` programmer's guide for more details.

- **Added new multi-process communication channel.**

  Added a generic channel in EAL for multi-process (primary/secondary) communication. Consumers of this channel need to register an action with an action name to response a message received; the actions will be identified by the action name and executed in the context of a new dedicated thread for this channel. The list of new APIs:

  - `rte_mp_register` and `rte_mp_unregister` are for action (un)registration.
  - `rte_mp_sendmsg` is for sending a message without blocking for a response.
  - `rte_mp_request` is for sending a request message and will block until it gets a reply message which is sent from the peer by `rte_mp_reply`.

- **Added GRO support for VxLAN-tunneled packets.**

  Added GRO support for VxLAN-tunneled packets. Supported VxLAN packets must contain an outer IPv4 header and inner TCP/IPv4 headers. VxLAN GRO doesn't check if input packets have correct checksums and doesn't update checksums for output packets. Additionally, it assumes the packets are complete (i.e., `MF==0 && frag_off==0`), when IP fragmentation is possible (i.e., `DF==0`).

- **Increased default Rx and Tx ring size in sample applications.**

  Increased the default `RX_RING_SIZE` and `TX_RING_SIZE` to 1024 entries in testpmd and the sample applications to give better performance in the general case. The user should experiment with various Rx and Tx ring sizes for their specific application to get best performance.

- **Added new DPDK build system using the tools "meson" and "ninja" [EXPERIMENTAL].**

  Added support for building DPDK using `meson` and `ninja`, which gives additional features, such as automatic build-time configuration, over the current build system using `make`. For instructions on how to do a DPDK build using the new system, see the instructions in `doc/build-sdk-meson.txt`.

  Note: This new build system support is incomplete at this point and is added as experimental in this release. The existing build system using `make` is unaffected by these changes, and can continue to be used for this and subsequent releases until such time as it's deprecation is announced.

4.2 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
+  librte_bbdev.so.1
   librte_bitratestats.so.2
   librte_bus_dpaa.so.1
   librte_bus_fslmc.so.1
   librte_bus_pci.so.1
   librte_bus_vdev.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
   librte_cryptodev.so.4
   librte_distributor.so.1
   librte_eal.so.6
   librte_ethdev.so.8
   librte_eventdev.so.3
   librte_flow_classify.so.1
   librte_gro.so.1
   librte_gso.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
   librte_latencystats.so.1
   librte_lpm.so.2
   librte_mbuf.so.3
   librte_mempool.so.3
   librte_meter.so.1
   librte_metrics.so.1
   librte_net.so.1
   librte_pci.so.1
   librte_pdump.so.2
   librte_pipeline.so.3
   librte_pmd_bnxt.so.2
   librte_pmd_bond.so.2
   librte_pmd_i40e.so.2
   librte_pmd_ixgbe.so.2
   librte_pmd_ring.so.2
   librte_pmd_softnic.so.1
   librte_pmd_vhost.so.2
   librte_port.so.3
   librte_power.so.1
+  librte_rawdev.so.1
   librte_reorder.so.1
   librte_ring.so.1
   librte_sched.so.1
   librte_security.so.1
   librte_table.so.3
   librte_timer.so.1
   librte_vhost.so.3
```

4.3 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations

- CPU
  - Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
  - Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
  - Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
  - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
  - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
  - Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
  - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
  - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz
  - Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
  - Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz
- OS:
  - CentOS 7.2
  - Fedora 25
  - Fedora 26
  - Fedora 27
  - FreeBSD 11
  - Red Hat Enterprise Linux Server release 7.3
  - SUSE Enterprise Linux 12
  - Wind River Linux 8
  - Ubuntu 14.04
  - Ubuntu 16.04
  - Ubuntu 16.10
  - Ubuntu 17.10
- NICs:
  - Intel(R) 82599ES 10 Gigabit Ethernet Controller
    - Firmware version: 0x61bf0001
    - Device id (pf/vf): 8086:10fb / 8086:10ed
    - Driver version: 5.2.3 (ixgbe)
  - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
    - Firmware version: 0x800003e7
    - Device id (pf/vf): 8086:15ad / 8086:15a8
    - Driver version: 4.4.6 (ixgbe)
  - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)

        - Firmware version: 6.01 0x80003221
        - Device id (pf/vf): 8086:1572 / 8086:154c
        - Driver version: 2.4.3 (i40e)
      - Intel Corporation Ethernet Connection X722 for 10GBASE-T
        - firmware-version: 6.01 0x80003221
        - Device id: 8086:37d2 / 8086:154c
        - Driver version: 2.4.3 (i40e)
      - Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
        - Firmware version: 6.01 0x80003221
        - Device id (pf/vf): 8086:158b / 8086:154c
        - Driver version: 2.4.3 (i40e)
      - Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
        - Firmware version: 6.01 0x8000321c
        - Device id (pf/vf): 8086:1583 / 8086:154c
        - Driver version: 2.4.3 (i40e)
      - Intel(R) Corporation I350 Gigabit Network Connection
        - Firmware version: 1.63, 0x80000dda
        - Device id (pf/vf): 8086:1521 / 8086:1520
        - Driver version: 5.3.0-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
  - CPU:
    - Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
    - Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
    - Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
  - OS:
    - Red Hat Enterprise Linux Server release 7.5 Beta (Maipo)
    - Red Hat Enterprise Linux Server release 7.4 (Maipo)
    - Red Hat Enterprise Linux Server release 7.3 (Maipo)
    - Red Hat Enterprise Linux Server release 7.2 (Maipo)
    - Ubuntu 17.10
    - Ubuntu 16.10

    - Ubuntu 16.04
- MLNX_OFED: 4.2-1.0.0.0
- MLNX_OFED: 4.3-0.1.6.0
- NICs:
    - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1007
        - Firmware version: 2.42.5000
    - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
        - Host interface: PCI Express 3.0 x8
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
        - Host interface: PCI Express 3.0 x16
        - Device ID: 15b3:1013
        - Firmware version: 12.21.1000 and above
    - Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
        - Host interface: PCI Express 3.0 x8

      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000 and above
    * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000 and above
    * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1017
      - Firmware version: 16.21.1000 and above
    * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
      - Host interface: PCI Express 4.0 x16
      - Device ID: 15b3:1019

            - Firmware version: 16.21.1000 and above

- ARM platforms with Mellanox(R) NICs combinations
  - CPU:
    - Qualcomm ARM 1.1 2500MHz
  - OS:
    - Ubuntu 16.04
  - MLNX_OFED: 4.2-1.0.0.0
  - NICs:
    - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1015
      - Firmware version: 14.21.1000
    - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1017
      - Firmware version: 16.21.1000

CHAPTER

FIVE

DPDK RELEASE 17.11

5.1 New Features

- **Extended port_id range from uint8_t to uint16_t.**

  Increased the `port_id` range from 8 bits to 16 bits in order to support more than 256 ports in DPDK. All ethdev APIs which have `port_id` as parameter have been changed.

- **Modified the return type of rte_eth_stats_reset.**

  Changed return type of `rte_eth_stats_reset` from `void` to `int` so that the caller can determine whether a device supports the operation or not and if the operation was carried out.

- **Added a new driver for Marvell Armada 7k/8k devices.**

  Added the new `mrvl` net driver for Marvell Armada 7k/8k devices. See the `../nics/mvpp2` NIC guide for more details on this new driver.

- **Updated mlx4 driver.**

  Updated the mlx4 driver including the following changes:

  – Isolated mode (rte_flow) can now be enabled anytime, not only during initial device configuration.

  – Flow rules now support up to 4096 priority levels usable at will by applications.

  – Enhanced error message to help debugging invalid/unsupported flow rules.

  – Flow rules matching all multicast and promiscuous traffic are now allowed.

  – No more software restrictions on flow rules with the RSS action, their configuration is much more flexible.

  – Significantly reduced memory footprint for Rx and Tx queue objects.

  – While supported, UDP RSS is temporarily disabled due to a remaining issue with its support in the Linux kernel.

  – The new RSS implementation does not automatically spread traffic according to the inner packet of VXLAN frames anymore, only the outer one (like other PMDs).

  – Partial (Tx only) support for secondary processes was broken and had to be removed.

  - Refactored driver to get rid of dependency on the components provided by Mellanox OFED and instead rely on the current and public rdma-core package and Linux version from now on.
  - Removed compile-time limitation on number of device instances the PMD can support.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  - Enabled the PMD to run on top of upstream Linux kernel and rdma-core libs, removing the dependency on specific Mellanox OFED libraries.
  - Improved PMD latency performance.
  - Improved PMD memory footprint.
  - Added support for vectorized Rx/Tx burst for ARMv8.
  - Added support for secondary process.
  - Added support for flow counters.
  - Added support for Rx hardware timestamp offload.
  - Added support for device removal event.

- **Added SoftNIC PMD.**

  Added a new SoftNIC PMD. This virtual device provides applications with software fallback support for traffic management.

- **Added support for NXP DPAA Devices.**

  Added support for NXP's DPAA devices - LS104x series. This includes:

  - DPAA Bus driver
  - DPAA Mempool driver for supporting offloaded packet memory pool
  - DPAA PMD for DPAA devices

  See the `../nics/dpaa` document for more details of this new driver.

- **Updated support for Cavium OCTEONTX Device.**

  Updated support for Cavium's OCTEONTX device (CN83xx). This includes:

  - OCTEONTX Mempool driver for supporting offloaded packet memory pool
  - OCTEONTX Ethdev PMD
  - OCTEONTX Eventdev-Ethdev Rx adapter

  See the `../nics/octeontx` document for more details of this new driver.

- **Added PF support to the Netronome NFP PMD.**

  Added PF support to the Netronome NFP PMD. Previously the NFP PMD only supported VFs. PF support is just as a basic DPDK port and has no VF management yet.

  PF support comes with firmware upload support which allows the PMD to independently work from kernel netdev NFP drivers.

  NFP 4000 devices are also now supported along with previous 6000 devices.

- **Updated bnxt PMD.**

  Major enhancements include:

  – Support for Flow API

  – Support for Tx and Rx descriptor status functions

- **Added bus agnostic functions to cryptodev for PMD initialization**

  Added new PMD assist, bus independent, functions `rte_cryptodev_pmd_parse_input_args()`, `rte_cryptodev_pmd_create()` and `rte_cryptodev_pmd_destroy()` for drivers to manage creation and destruction of new device instances.

- **Updated QAT crypto PMD.**

  Added several performance enhancements:

  – Removed atomics from the internal queue pair structure.

  – Added coalesce writes to HEAD CSR on response processing.

  – Added coalesce writes to TAIL CSR on request processing.

  In addition support was added for the AES CCM algorithm.

- **Updated the AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for:

  – The DES CBC algorithm.

  – The DES DOCSIS BPI algorithm.

  This change requires version 0.47 of the IPSec Multi-buffer library. For more details see the `../cryptodevs/aesni_mb` documentation.

- **Updated the OpenSSL PMD.**

  The OpenSSL PMD has been updated with additional support for:

  – The DES CBC algorithm.

  – The AES CCM algorithm.

- **Added NXP DPAA SEC crypto PMD.**

  A new `dpaa_sec` hardware based crypto PMD for NXP DPAA devices has been added. See the `../cryptodevs/dpaa_sec` document for more details.

- **Added MRVL crypto PMD.**

  A new crypto PMD has been added, which provides several ciphering and hashing algorithms. All cryptography operations use the MUSDK library crypto API. See the `../cryptodevs/mvsam` document for more details.

- **Add new benchmarking mode to dpdk-test-crypto-perf application.**

  Added a new "PMD cyclecount" benchmark mode to the `dpdk-test-crypto-perf` application to display a detailed breakdown of CPU cycles used by hardware acceleration.

- **Added the Security Offload Library.**

Added an experimental library - `rte_security`. This provide security APIs for protocols like IPSec using inline ipsec offload to ethernet devices or full protocol offload with lookaside crypto devices.

See the `../prog_guide/rte_security` section of the DPDK Programmers Guide document for more information.

- **Updated the DPAA2_SEC crypto driver to support rte_security.**

  Updated the `dpaa2_sec` crypto PMD to support `rte_security` lookaside protocol offload for IPSec.

- **Updated the IXGBE ethernet driver to support rte_security.**

  Updated ixgbe ethernet PMD to support `rte_security` inline IPSec offload.

- **Updated i40e driver to support GTP-C/GTP-U.**

  Updated i40e PMD to support GTP-C/GTP-U with GTP-C/GTP-U supporting profiles which can be programmed by dynamic device personalization (DDP) process.

- **Added the i40e ethernet driver to support queue region feature.**

  This feature enable queue regions configuration for RSS in PF, so that different traffic classes or different packet classification types can be separated into different queues in different queue regions.

- **Updated ipsec-secgw application to support rte_security.**

  Updated the `ipsec-secgw` sample application to support `rte_security` actions for ipsec inline and full protocol offload using lookaside crypto offload.

- **Added IOMMU support to libvhost-user**

  Implemented device IOTLB in the Vhost-user backend, and enabled Virtio's IOMMU feature. The feature is disabled by default, and can be enabled by setting `RTE_VHOST_USER_IOMMU_SUPPORT` flag at vhost device registration time.

- **Added the Event Ethernet Adapter Library.**

  Added the Event Ethernet Adapter library. This library provides APIs for eventdev applications to configure the ethdev for eventdev packet flow.

- **Updated DPAA2 Event PMD for the Event Ethernet Adapter.**

  Added support for the eventdev ethernet adapter for DPAA2.

- **Added Membership library (rte_member).**

  Added a new data structure library called the Membership Library.

  The Membership Library is an extension and generalization of a traditional filter (for example Bloom Filter) structure that has multiple usages in a wide variety of workloads and applications. In general, the Membership Library is a data structure that provides a "set-summary" and responds to set-membership queries whether a certain member belongs to a set(s).

  The library provides APIs for DPDK applications to insert a new member, delete an existing member, and query the existence of a member in a given set, or a group of sets. For the case of a group of sets the library will return not only whether the element has been inserted in one of the sets but also which set it belongs to.

See the `../prog_guide/member_lib` documentation in the Programmers Guide, for more information.

- **Added the Generic Segmentation Offload Library.**

  Added the Generic Segmentation Offload (GSO) library to enable applications to split large packets (e.g. MTU is 64KB) into small ones (e.g. MTU is 1500B). Supported packet types are:

  – TCP/IPv4 packets.

  – VxLAN packets, which must have an outer IPv4 header, and contain an inner TCP/IPv4 packet.

  – GRE packets, which must contain an outer IPv4 header, and inner TCP/IPv4 headers.

  The GSO library doesn't check if the input packets have correct checksums, and doesn't update checksums for output packets. Additionally, the GSO library doesn't process IP fragmented packets.

- **Added the Flow Classification Library.**

  Added an experimental Flow Classification library to provide APIs for DPDK applications to classify an input packet by matching it against a set of flow rules. It uses the `librte_table` API to manage the flow rules.

5.2 Resolved Issues

- **Service core fails to call service callback due to atomic lock**

  In a specific configuration of multi-thread unsafe services and service cores, a service core previously did not correctly release the atomic lock on the service. This would result in the cores polling the service, but it looked like another thread was executing the service callback. The logic for atomic locking of the services has been fixed and refactored for readability.

5.3 API Changes

- **Ethdev device name length increased.**

  The size of internal device name has been increased to 64 characters to allow for storing longer bus specific names.

- **Removed the Ethdev RTE_ETH_DEV_DETACHABLE flag.**

  Removed the Ethdev `RTE_ETH_DEV_DETACHABLE` flag. This flag is not required anymore, with the new hotplug implementation. It has been removed from the ether library. Its semantics are now expressed at the bus and PMD level.

- **Service cores API updated for usability**

  The service cores API has been changed, removing pointers from the API where possible, and instead using integer IDs to identify each service. This simplifies application code, aids debugging, and provides better encapsulation. A summary of the main changes made is as follows:

  - Services identified by ID not by `rte_service_spec` pointer
  - Reduced API surface by using `set` functions instead of enable/disable
  - Reworked `rte_service_register` to provide the service ID to registrar
  - Reworked start and stop APIs into `rte_service_runstate_set`
  - Added API to set runstate of service implementation to indicate readiness

- **The following changes have been made in the mempool library**
  - Moved `flags` datatype from `int` to `unsigned int` for `rte_mempool`.
  - Removed `__rte_unused int flag` param from `rte_mempool_generic_put` and `rte_mempool_generic_get` API.
  - Added `flags` param in `rte_mempool_xmem_size` and `rte_mempool_xmem_usage`.
  - `rte_mem_phy2mch` was used in Xen dom0 to obtain the physical address; remove this API as Xen dom0 support was removed.

- **Added IOVA aliases related to physical address handling.**

  Some data types, structure members and functions related to physical address handling are deprecated and have new aliases with IOVA wording. For example:

  - `phys_addr_t` can be often replaced by `rte_iova_t` of same size.
  - `RTE_BAD_PHYS_ADDR` is often replaced by `RTE_BAD_IOVA` of same value.
  - `rte_memseg.phys_addr` is aliased with `rte_memseg.iova_addr`.
  - `rte_mem_virt2phy()` can often be replaced by `rte_mem_virt2iova`.
  - `rte_malloc_virt2phy` is aliased with `rte_malloc_virt2iova`.
  - `rte_memzone.phys_addr` is aliased with `rte_memzone.iova`.
  - `rte_mempool_objhdr.physaddr` is aliased with `rte_mempool_objhdr.iova`.
  - `rte_mempool_memhdr.phys_addr` is aliased with `rte_mempool_memhdr.iova`.
  - `rte_mempool_virt2phy()` can be replaced by `rte_mempool_virt2iova()`.
  - `rte_mempool_populate_phys*()` are aliased with `rte_mempool_populate_iova*()`
  - `rte_mbuf.buf_physaddr` is aliased with `rte_mbuf.buf_iova`.
  - `rte_mbuf_data_dma_addr*()` are aliased with `rte_mbuf_data_iova*()`.
  - `rte_pktmbuf_mtophys*` are aliased with `rte_pktmbuf_iova*()`.

- **PCI bus API moved outside of the EAL**

  The PCI bus previously implemented within the EAL has been moved. A first part has been added as an RTE library providing PCI helpers to parse device locations or other such utilities. A second part consisting of the actual bus driver has been moved to its proper subdirectory, without changing its functionalities.

  As such, several PCI-related functions are not exposed by the EAL anymore:

  - rte_pci_detach
  - rte_pci_dump
  - rte_pci_ioport_map
  - rte_pci_ioport_read
  - rte_pci_ioport_unmap
  - rte_pci_ioport_write
  - rte_pci_map_device
  - rte_pci_probe
  - rte_pci_probe_one
  - rte_pci_read_config
  - rte_pci_register
  - rte_pci_scan
  - rte_pci_unmap_device
  - rte_pci_unregister
  - rte_pci_write_config

  These functions are made available either as part of librte_pci or librte_bus_pci.

- **Moved vdev bus APIs outside of the EAL**

  Moved the following APIs from librte_eal to librte_bus_vdev:

  - rte_vdev_init
  - rte_vdev_register
  - rte_vdev_uninit
  - rte_vdev_unregister

- **Add return value to stats_get dev op API**

  The stats_get dev op API return value has been changed to be int. In this way PMDs can return an error value in case of failure at stats getting process time.

- **Modified the rte_cryptodev_allocate_driver function.**

  Modified the rte_cryptodev_allocate_driver() function in the cryptodev library. An extra parameter struct cryptodev_driver *crypto_drv has been added.

- **Removed virtual device bus specific functions from librte_cryptodev.**

  The functions rte_cryptodev_vdev_parse_init_params() and rte_cryptodev_vdev_pmd_init() have been removed from librte_cryptodev and have been replaced by non bus specific functions rte_cryptodev_pmd_parse_input_args() and rte_cryptodev_pmd_create().

  The rte_cryptodev_create_vdev() function was removed to avoid the dependency on vdev in librte_cryptodev; instead, users can call rte_vdev_init() directly.

- **Removed PCI device bus specific functions from librte_cryptodev.**

  The functions `rte_cryptodev_pci_generic_probe()` and `rte_cryptodev_pci_generic_remove()` have been removed from librte_cryptodev and have been replaced by non bus specific functions `rte_cryptodev_pmd_create()` and `rte_cryptodev_pmd_destroy()`.

- **Removed deprecated functions to manage log level or type.**

  The functions `rte_set_log_level()`, `rte_get_log_level()`, `rte_set_log_type()` and `rte_get_log_type()` have been removed.

  They are respectively replaced by `rte_log_set_global_level()`, `rte_log_get_global_level()`, `rte_log_set_level()` and `rte_log_get_level()`.

- **Removed mbuf flags PKT_RX_VLAN_PKT and PKT_RX_QINQ_PKT.**

  The `mbuf` flags `PKT_RX_VLAN_PKT` and `PKT_RX_QINQ_PKT` have been removed since their behavior was not properly described.

- **Added mbuf flags PKT_RX_VLAN and PKT_RX_QINQ.**

  Two `mbuf` flags have been added to indicate that the VLAN identifier has been saved in in the `mbuf` structure. For instance:

  - If VLAN is not stripped and TCI is saved: `PKT_RX_VLAN`
  - If VLAN is stripped and TCI is saved: `PKT_RX_VLAN | PKT_RX_VLAN_STRIPPED`

- **Modified the vlan_offload_set_t function prototype in the ethdev library.**

  Modified the `vlan_offload_set_t` function prototype in the ethdev library. The return value has been changed from `void` to `int` so the caller can determine whether the backing device supports the operation or if the operation was successfully performed.

5.4 ABI Changes

- **Extended port_id range.**

  The size of the field `port_id` in the `rte_eth_dev_data` structure has changed, as described in the *New Features* section above.

- **New parameter added to rte_eth_dev.**

  A new parameter `security_ctx` has been added to `rte_eth_dev` to support security operations like IPSec inline.

- **New parameter added to rte_cryptodev.**

  A new parameter `security_ctx` has been added to `rte_cryptodev` to support security operations like lookaside crypto.

5.5 Removed Items

- Xen dom0 in EAL has been removed, as well as the xenvirt PMD and vhost_xen.

- The crypto performance unit tests have been removed, replaced by the `dpdk-test-crypto-perf` application.

5.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
+  librte_bitratestats.so.2
+  librte_bus_dpaa.so.1
+  librte_bus_fslmc.so.1
+  librte_bus_pci.so.1
+  librte_bus_vdev.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
+  librte_cryptodev.so.4
   librte_distributor.so.1
+  librte_eal.so.6
+  librte_ethdev.so.8
+  librte_eventdev.so.3
+  librte_flow_classify.so.1
   librte_gro.so.1
+  librte_gso.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
   librte_latencystats.so.1
   librte_lpm.so.2
   librte_mbuf.so.3
+  librte_mempool.so.3
   librte_meter.so.1
   librte_metrics.so.1
   librte_net.so.1
+  librte_pci.so.1
+  librte_pdump.so.2
   librte_pipeline.so.3
+  librte_pmd_bnxt.so.2
+  librte_pmd_bond.so.2
+  librte_pmd_i40e.so.2
+  librte_pmd_ixgbe.so.2
   librte_pmd_ring.so.2
+  librte_pmd_softnic.so.1
+  librte_pmd_vhost.so.2
   librte_port.so.3
   librte_power.so.1
   librte_reorder.so.1
   librte_ring.so.1
   librte_sched.so.1
+  librte_security.so.1
+  librte_table.so.3
   librte_timer.so.1
   librte_vhost.so.3
```

5.7 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations

- CPU
  - Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
  - Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
  - Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
  - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
  - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
  - Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
  - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
  - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz
  - Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
- OS:
  - CentOS 7.2
  - Fedora 25
  - Fedora 26
  - FreeBSD 11
  - Red Hat Enterprise Linux Server release 7.3
  - SUSE Enterprise Linux 12
  - Wind River Linux 8
  - Ubuntu 16.04
  - Ubuntu 16.10
- NICs:
  - Intel(R) 82599ES 10 Gigabit Ethernet Controller
    - Firmware version: 0x61bf0001
    - Device id (pf/vf): 8086:10fb / 8086:10ed
    - Driver version: 5.2.3 (ixgbe)
  - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
    - Firmware version: 0x800003e7
    - Device id (pf/vf): 8086:15ad / 8086:15a8
    - Driver version: 4.4.6 (ixgbe)
  - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
    - Firmware version: 6.01 0x80003205
    - Device id (pf/vf): 8086:1572 / 8086:154c
    - Driver version: 2.1.26 (i40e)
  - Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)

            - Firmware version: 6.01 0x80003204
            - Device id (pf/vf): 8086:1572 / 8086:154c
            - Driver version: 2.1.26 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
            - Firmware version: 6.01 0x80003221
            - Device id (pf/vf): 8086:158b
            - Driver version: 2.1.26 (i40e)
        * Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
            - Firmware version: 6.01 0x8000321c
            - Device id (pf/vf): 8086:1583 / 8086:154c
            - Driver version: 2.1.26 (i40e)
        * Intel(R) Corporation I350 Gigabit Network Connection
            - Firmware version: 1.63, 0x80000dda
            - Device id (pf/vf): 8086:1521 / 8086:1520
            - Driver version: 5.3.0-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
    - Platform details:
        * Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
        * Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
        * Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
        * Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
        * Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
    - OS:
        * Red Hat Enterprise Linux Server release 7.3 (Maipo)
        * Red Hat Enterprise Linux Server release 7.2 (Maipo)
        * Ubuntu 16.10
        * Ubuntu 16.04
        * Ubuntu 14.04
    - MLNX_OFED: 4.2-1.0.0.0
    - NICs:
        * Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1007

      - Firmware version: 2.42.5000
    * Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.21.1000
    * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)

            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.21.1000
        * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.21.1000
        * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.21.1000
        * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.21.1000
        * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.21.1000
        * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.21.1000
        * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.21.1000
- ARM platforms with Mellanox(R) NICs combinations
    - Platform details:
        * Qualcomm ARM 1.1 2500MHz
    - OS:
        * Ubuntu 16.04
    - MLNX_OFED: 4.2-1.0.0.0
    - NICs:

* Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
  - Host interface: PCI Express 3.0 x8
  - Device ID: 15b3:1015
  - Firmware version: 14.21.1000
* Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
  - Host interface: PCI Express 3.0 x16
  - Device ID: 15b3:1017
  - Firmware version: 16.21.1000

CHAPTER

SIX

DPDK RELEASE 17.08

6.1 New Features

- **Increase minimum x86 ISA version to SSE4.2.**

  Starting with version 17.08, DPDK requires SSE4.2 to run on x86. Previous versions required SSE3.

- **Added Service Core functionality.**

  The service core functionality added to EAL allows DPDK to run services such as software PMDs on lcores without the application manually running them. The service core infrastructure allows flexibility of running multiple services on the same service lcore, and provides the application with powerful APIs to configure the mapping from service lcores to services.

- **Added Generic Receive Offload API.**

  Added Generic Receive Offload (GRO) API support to reassemble TCP/IPv4 packets. The GRO API assumes all input packets have the correct checksums. GRO API doesn't update checksums for merged packets. If input packets are IP fragmented, the GRO API assumes they are complete packets (i.e. with L4 headers).

- **Added Fail-Safe PMD**

  Added the new Fail-Safe PMD. This virtual device allows applications to support seamless hotplug of devices. See the `/nics/fail_safe` guide for more details about this driver.

- **Added support for generic flow API (rte_flow) on igb NICs.**

  This API provides a generic means of configuring hardware to match specific ingress or egress traffic, altering its behavior and querying related counters according to any number of user-defined rules.

  Added generic flow API support for Ethernet, IPv4, UDP, TCP and RAW pattern items with QUEUE actions. There are four types of filter support for this feature on igb.

- **Added support for generic flow API (rte_flow) on enic.**

  Added flow API support for outer Ethernet, VLAN, IPv4, IPv6, UDP, TCP, SCTP, VxLAN and inner Ethernet, VLAN, IPv4, IPv6, UDP and TCP pattern items with QUEUE, MARK, FLAG and VOID actions for ingress traffic.

- **Added support for Chelsio T6 family of adapters**

  The CXGBE PMD was updated to run Chelsio T6 family of adapters.

- **Added latency and performance improvements for cxgbe**

  the Tx and Rx path in cxgbe were reworked to improve performance. In addition the latency was reduced for slow traffic.

- **Updated the bnxt PMD.**

  Updated the bnxt PMD. The major enhancements include:

  - Support MTU modification.
  - Add support for LRO.
  - Add support for VLAN filter and strip functionality.
  - Additional enhancements to add support for more dev_ops.
  - Added PMD specific APIs mainly to control VF from PF.
  - Update HWRM version to 1.7.7

- **Added support for Rx interrupts on mlx4 driver.**

  Rx queues can be now be armed with an interrupt which will trigger on the next packet arrival.

- **Updated mlx5 driver.**

  Updated the mlx5 driver including the following changes:

  - Added vectorized Rx/Tx burst for x86.
  - Added support for isolated mode from flow API.
  - Reworked the flow drop action to implement in hardware classifier.
  - Improved Rx interrupts management.

- **Updated szedata2 PMD.**

  Added support for firmware with multiple Ethernet ports per physical port.

- **Updated dpaa2 PMD.**

  Updated dpaa2 PMD. Major enhancements include:

  - Added support for MAC Filter configuration.
  - Added support for Segmented Buffers.
  - Added support for VLAN filter and strip functionality.
  - Additional enhancements to add support for more dev_ops.
  - Optimized the packet receive path

- **Reorganized the symmetric crypto operation structure.**

  The crypto operation (`rte_crypto_sym_op`) has been reorganized as follows:

  - Removed the `rte_crypto_sym_op_sess_type` field.
  - Replaced the pointer and physical address of IV with offset from the start of the crypto operation.
  - Moved length and offset of cipher IV to `rte_crypto_cipher_xform`.

  - Removed “Additional Authentication Data” (AAD) length.
  - Removed digest length.
  - Removed AAD pointer and physical address from `auth` structure.
  - Added `aead` structure, containing parameters for AEAD algorithms.

- **Reorganized the crypto operation structure.**

  The crypto operation (`rte_crypto_op`) has been reorganized as follows:

  - Added the `rte_crypto_op_sess_type` field.
  - The enumerations `rte_crypto_op_status` and `rte_crypto_op_type` have been modified to be `uint8_t` values.
  - Removed the field `opaque_data`.
  - Pointer to `rte_crypto_sym_op` has been replaced with a zero length array.

- **Reorganized the crypto symmetric session structure.**

  The crypto symmetric session structure (`rte_cryptodev_sym_session`) has been reorganized as follows:

  - The `dev_id` field has been removed.
  - The `driver_id` field has been removed.
  - The mempool pointer `mp` has been removed.
  - Replaced `private` marker with array of pointers to private data sessions `sess_private_data`.

- **Updated cryptodev library.**

  - Added AEAD algorithm specific functions and structures, so it is not necessary to use a combination of cipher and authentication structures anymore.
  - Added helper functions for crypto device driver identification.
  - Added support for multi-device sessions, so a single session can be used in multiple drivers.
  - Added functions to initialize and free individual driver private data with the same session.

- **Updated dpaa2_sec crypto PMD.**

  Added support for AES-GCM and AES-CTR.

- **Updated the AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for:

  - 12-byte IV on AES Counter Mode, apart from the previous 16-byte IV.

- **Updated the AES-NI GCM PMD.**

  The AES-NI GCM PMD was migrated from the ISA-L library to the Multi Buffer library, as the latter library has Scatter Gather List support now. The migration entailed adding additional support for 192-bit keys.

- **Updated the Cryptodev Scheduler PMD.**

  Added a multicore based distribution mode, which distributes the enqueued crypto operations among several slaves, running on different logical cores.

- **Added NXP DPAA2 Eventdev PMD.**

  Added the new dpaa2 eventdev driver for NXP DPAA2 devices. See the "Event Device Drivers" document for more details on this new driver.

- **Added dpdk-test-eventdev test application.**

  The dpdk-test-eventdev tool is a Data Plane Development Kit (DPDK) application that allows exercising various eventdev use cases. This application has a generic framework to add new eventdev based test cases to verify functionality and measure the performance parameters of DPDK eventdev devices.

6.2 Known Issues

- **Starting with version 17.08, libnuma is required to build DPDK.**

6.3 API Changes

- **Modified the _rte_eth_dev_callback_process function in the ethdev library.**

  The function `_rte_eth_dev_callback_process()` has been modified. The return value has been changed from void to int and an extra parameter `void *ret_param` has been added.

- **Moved bypass functions from the rte_ethdev library to ixgbe PMD**
  - The following rte_ethdev library functions were removed:
    * `rte_eth_dev_bypass_event_show()`
    * `rte_eth_dev_bypass_event_store()`
    * `rte_eth_dev_bypass_init()`
    * `rte_eth_dev_bypass_state_set()`
    * `rte_eth_dev_bypass_state_show()`
    * `rte_eth_dev_bypass_ver_show()`
    * `rte_eth_dev_bypass_wd_reset()`
    * `rte_eth_dev_bypass_wd_timeout_show()`
    * `rte_eth_dev_wd_timeout_store()`
  - The following ixgbe PMD functions were added:
    * `rte_pmd_ixgbe_bypass_event_show()`
    * `rte_pmd_ixgbe_bypass_event_store()`
    * `rte_pmd_ixgbe_bypass_init()`
    * `rte_pmd_ixgbe_bypass_state_set()`

  * `rte_pmd_ixgbe_bypass_state_show()`
  * `rte_pmd_ixgbe_bypass_ver_show()`
  * `rte_pmd_ixgbe_bypass_wd_reset()`
  * `rte_pmd_ixgbe_bypass_wd_timeout_show()`
  * `rte_pmd_ixgbe_bypass_wd_timeout_store()`

- **Reworked rte_cryptodev library.**

  The rte_cryptodev library has been reworked and updated. The following changes have been made to it:

  – The crypto device type enumeration has been removed from cryptodev library.

  – The function `rte_crypto_count_devtype()` has been removed, and replaced by the new function `rte_crypto_count_by_driver()`.

  – Moved crypto device driver names definitions to the particular PMDs. These names are not public anymore.

  – The `rte_cryptodev_configure()` function does not create the session mempool for the device anymore.

  – The `rte_cryptodev_queue_pair_attach_sym_session()` and `rte_cryptodev_queue_pair_dettach_sym_session()` functions require the new parameter `device id`.

  – Parameters of `rte_cryptodev_sym_session_create()` were modified to accept `mempool`, instead of `device id` and `rte_crypto_sym_xform`.

  – Removed `device id` parameter from `rte_cryptodev_sym_session_free()`.

  – Added a new field `session_pool` to `rte_cryptodev_queue_pair_setup()`.

  – Removed `aad_size` parameter from `rte_cryptodev_sym_capability_check_auth()`.

  – Added `iv_size` parameter to `rte_cryptodev_sym_capability_check_auth()`.

  – Removed `RTE_CRYPTO_OP_STATUS_ENQUEUED` from enum `rte_crypto_op_status`.

6.4 ABI Changes

- Changed type of `domain` field in `rte_pci_addr` to `uint32_t` to follow the PCI standard.
- Added new `rte_bus` experimental APIs available as operators within the `rte_bus` structure.
- Made `rte_devargs` structure internal device representation generic to prepare for a bus-agnostic EAL.
- **Reorganized the crypto operation structures.**

  Some fields have been modified in the `rte_crypto_op` and `rte_crypto_sym_op` structures, as described in the *New Features* section.

- **Reorganized the crypto symmetric session structure.**

  Some fields have been modified in the `rte_cryptodev_sym_session` structure, as described in the *New Features* section.

- **Reorganized the rte_crypto_sym_cipher_xform structure.**
  - Added cipher IV length and offset parameters.
  - Changed field size of key length from `size_t` to `uint16_t`.
- **Reorganized the rte_crypto_sym_auth_xform structure.**
  - Added authentication IV length and offset parameters.
  - Changed field size of AAD length from `uint32_t` to `uint16_t`.
  - Changed field size of digest length from `uint32_t` to `uint16_t`.
  - Removed AAD length.
  - Changed field size of key length from `size_t` to `uint16_t`.
- Replaced `dev_type` enumeration with `uint8_t driver_id` in `rte_cryptodev_info` and `rte_cryptodev` structures.
- Removed `session_mp` from `rte_cryptodev_config`.

6.5 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
   librte_bitratestats.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
+  librte_cryptodev.so.3
   librte_distributor.so.1
+  librte_eal.so.5
+  librte_ethdev.so.7
+  librte_eventdev.so.2
+  librte_gro.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
   librte_latencystats.so.1
   librte_lpm.so.2
   librte_mbuf.so.3
   librte_mempool.so.2
   librte_meter.so.1
   librte_metrics.so.1
   librte_net.so.1
   librte_pdump.so.1
   librte_pipeline.so.3
   librte_pmd_bond.so.1
   librte_pmd_ring.so.2
   librte_port.so.3
   librte_power.so.1
   librte_reorder.so.1
   librte_ring.so.1
```

```
librte_sched.so.1
librte_table.so.2
librte_timer.so.1
librte_vhost.so.3
```

6.6 Tested Platforms

- Intel(R) platforms with Mellanox(R) NICs combinations
  - Platform details:
    - Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
  - OS:
    - Red Hat Enterprise Linux Server release 7.3 (Maipo)
    - Red Hat Enterprise Linux Server release 7.2 (Maipo)
    - Ubuntu 16.10
    - Ubuntu 16.04
    - Ubuntu 14.04
  - MLNX_OFED: 4.1-1.0.2.0
  - NICs:
    - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1007
      - Firmware version: 2.40.5030
    - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.2000
    - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.2000
    - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013

    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)

            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.2000
        * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.2000
        * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.19.1200
        * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.19.1200

- Intel(R) platforms with Intel(R) NICs combinations
    - CPU
        * Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
        * Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
        * Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
        * Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
        * Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
        * Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
        * Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
        * Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz
        * Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
    - OS:
        * CentOS 7.2
        * Fedora 25
        * FreeBSD 11
        * Red Hat Enterprise Linux Server release 7.3
        * SUSE Enterprise Linux 12
        * Wind River Linux 8
        * Ubuntu 16.04

    - Ubuntu 16.10

- NICs:
    - Intel(R) 82599ES 10 Gigabit Ethernet Controller
        - Firmware version: 0x61bf0001
        - Device id (pf/vf): 8086:10fb / 8086:10ed
        - Driver version: 4.0.1-k (ixgbe)
    - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
        - Firmware version: 0x800001cf
        - Device id (pf/vf): 8086:15ad / 8086:15a8
        - Driver version: 4.2.5 (ixgbe)
    - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
        - Firmware version: 6.01 0x80003205
        - Device id (pf/vf): 8086:1572 / 8086:154c
        - Driver version: 2.0.19 (i40e)
    - Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
        - Firmware version: 6.01 0x80003204
        - Device id (pf/vf): 8086:1572 / 8086:154c
        - Driver version: 2.0.19 (i40e)
    - Intel(R) Ethernet Converged Network Adapter XXV710-DA2 (2x25G)
        - Firmware version: 6.01 0x80003221
        - Device id (pf/vf): 8086:158b
        - Driver version: 2.0.19 (i40e)
    - Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
        - Firmware version: 6.01 0x8000321c
        - Device id (pf/vf): 8086:1583 / 8086:154c
        - Driver version: 2.0.19 (i40e)
    - Intel(R) Corporation I350 Gigabit Network Connection
        - Firmware version: 1.48, 0x800006e7
        - Device id (pf/vf): 8086:1521 / 8086:1520
        - Driver version: 5.2.13-k (igb)

CHAPTER

SEVEN

DPDK RELEASE 17.05

7.1 New Features

- **Reorganized mbuf structure.**

  The mbuf structure has been reorganized as follows:

  - Align fields to facilitate the writing of `data_off`, `refcnt`, and `nb_segs` in one operation.
  - Use 2 bytes for port and number of segments.
  - Move the sequence number to the second cache line.
  - Add a timestamp field.
  - Set default value for `refcnt`, `next` and `nb_segs` at mbuf free.

- **Added mbuf raw free API.**

  Moved `rte_mbuf_raw_free()` and `rte_pktmbuf_prefree_seg()` functions to the public API.

- **Added free Tx mbuf on demand API.**

  Added a new function `rte_eth_tx_done_cleanup()` which allows an application to request the driver to release mbufs that are no longer in use from a Tx ring, independent of whether or not the `tx_rs_thresh` has been crossed.

- **Added device removal interrupt.**

  Added a new ethdev event `RTE_ETH_DEV_INTR_RMV` to signify the sudden removal of a device. This event can be advertised by PCI drivers and enabled accordingly.

- **Added EAL dynamic log framework.**

  Added new APIs to dynamically register named log types, and control the level of each type independently.

- **Added descriptor status ethdev API.**

  Added a new API to get the status of a descriptor.

  For Rx, it is almost similar to the `rx_descriptor_done` API, except it differentiates descriptors which are held by the driver and not returned to the hardware. For Tx, it is a new API.

- **Increased number of next hops for LPM IPv6 to 2^21.**

  The next_hop field has been extended from 8 bits to 21 bits for IPv6.

- **Added VFIO hotplug support.**

  Added hotplug support for VFIO in addition to the existing UIO support.

- **Added PowerPC support to pci probing for vfio-pci devices.**

  Enabled sPAPR IOMMU based pci probing for vfio-pci devices.

- **Kept consistent PMD batching behavior.**

  Removed the limit of fm10k/i40e/ixgbe Tx burst size and vhost Rx/Tx burst size in order to support the same policy of "make an best effort to Rx/Tx pkts" for PMDs.

- **Updated the ixgbe base driver.**

  Updated the ixgbe base driver, including the following changes:

  - Add link block check for KR.
  - Complete HW initialization even if SFP is not present.
  - Add VF xcast promiscuous mode.

- **Added PowerPC support for i40e and its vector PMD.**

  Enabled i40e PMD and its vector PMD by default in PowerPC.

- **Added VF max bandwidth setting in i40e.**

  Enabled capability to set the max bandwidth for a VF in i40e.

- **Added VF TC min and max bandwidth setting in i40e.**

  Enabled capability to set the min and max allocated bandwidth for a TC on a VF in i40.

- **Added TC strict priority mode setting on i40e.**

  There are 2 Tx scheduling modes supported for TCs by i40e HW: round robin mode and strict priority mode. By default the round robin mode is used. It is now possible to change the Tx scheduling mode for a TC. This is a global setting on a physical port.

- **Added i40e dynamic device personalization support.**

  - Added dynamic device personalization processing to i40e firmware.

- **Updated i40e driver to support MPLSoUDP/MPLSoGRE.**

  Updated i40e PMD to support MPLSoUDP/MPLSoGRE with MPLSoUDP/MPLSoGRE supporting profiles which can be programmed by dynamic device personalization (DDP) process.

- **Added Cloud Filter for QinQ steering to i40e.**

  - Added a QinQ cloud filter on the i40e PMD, for steering traffic to a VM using both VLAN tags. Note, this feature is not supported in Vector Mode.

- **Updated mlx5 PMD.**

  Updated the mlx5 driver, including the following changes:

  - Added Generic flow API support for classification according to ether type.

  - Extended Generic flow API support for classification of IPv6 flow according to Vtc flow, Protocol and Hop limit.
  - Added Generic flow API support for FLAG action.
  - Added Generic flow API support for RSS action.
  - Added support for TSO for non-tunneled and VXLAN packets.
  - Added support for hardware Tx checksum offloads for VXLAN packets.
  - Added support for user space Rx interrupt mode.
  - Improved ConnectX-5 single core and maximum performance.

- **Updated mlx4 PMD.**

  Updated the mlx4 driver, including the following changes:

  - Added support for Generic flow API basic flow items and actions.
  - Added support for device removal event.

- **Updated the sfc_efx driver.**

  - Added Generic Flow API support for Ethernet, VLAN, IPv4, IPv6, UDP and TCP pattern items with QUEUE action for ingress traffic.
  - Added support for virtual functions (VFs).

- **Added LiquidIO network PMD.**

  Added poll mode driver support for Cavium LiquidIO II server adapter VFs.

- **Added Atomic Rules Arkville PMD.**

  Added a new poll mode driver for the Arkville family of devices from Atomic Rules. The net/ark PMD supports line-rate agnostic, multi-queue data movement on Arkville core FPGA instances.

- **Added support for NXP DPAA2 - FSLMC bus.**

  Added the new bus “fslmc” driver for NXP DPAA2 devices. See the “Network Interface Controller Drivers” document for more details of this new driver.

- **Added support for NXP DPAA2 Network PMD.**

  Added the new “dpaa2” net driver for NXP DPAA2 devices. See the “Network Interface Controller Drivers” document for more details of this new driver.

- **Added support for the Wind River Systems AVP PMD.**

  Added a new networking driver for the AVP device type. Theses devices are specific to the Wind River Systems virtualization platforms.

- **Added vmxnet3 version 3 support.**

  Added support for vmxnet3 version 3 which includes several performance enhancements such as configurable Tx data ring, Receive Data Ring, and the ability to register memory regions.

- **Updated the TAP driver.**

  Updated the TAP PMD to:

  - Support MTU modification.
  - Support packet type for Rx.
  - Support segmented packets on Rx and Tx.
  - Speed up Rx on TAP when no packets are available.
  - Support capturing traffic from another netdevice.
  - Dynamically change link status when the underlying interface state changes.
  - Added Generic Flow API support for Ethernet, VLAN, IPv4, IPv6, UDP and TCP pattern items with DROP, QUEUE and PASSTHRU actions for ingress traffic.

- **Added MTU feature support to Virtio and Vhost.**

  Implemented new Virtio MTU feature in Vhost and Virtio:

  - Add `rte_vhost_mtu_get()` API to Vhost library.
  - Enable Vhost PMD's MTU get feature.
  - Get max MTU value from host in Virtio PMD

- **Added interrupt mode support for virtio-user.**

  Implemented Rxq interrupt mode and LSC support for virtio-user as a virtual device. Supported cases:

  - Rxq interrupt for virtio-user + vhost-user as the backend.
  - Rxq interrupt for virtio-user + vhost-kernel as the backend.
  - LSC interrupt for virtio-user + vhost-user as the backend.

- **Added event driven programming model library (rte_eventdev).**

  This API introduces an event driven programming model.

  In a polling model, lcores poll ethdev ports and associated Rx queues directly to look for a packet. By contrast in an event driven model, lcores call the scheduler that selects packets for them based on programmer-specified criteria. The Eventdev library adds support for an event driven programming model, which offers applications automatic multicore scaling, dynamic load balancing, pipelining, packet ingress order maintenance and synchronization services to simplify application packet processing.

  By introducing an event driven programming model, DPDK can support both polling and event driven programming models for packet processing, and applications are free to choose whatever model (or combination of the two) best suits their needs.

- **Added Software Eventdev PMD.**

  Added support for the software eventdev PMD. The software eventdev is a software based scheduler device that implements the eventdev API. This PMD allows an application to configure a pipeline using the eventdev library, and run the scheduling workload on a CPU core.

- **Added Cavium OCTEONTX Eventdev PMD.**

  Added the new octeontx ssovf eventdev driver for OCTEONTX devices. See the "Event Device Drivers" document for more details on this new driver.

- **Added information metrics library.**

  Added a library that allows information metrics to be added and updated by producers, typically other libraries, for later retrieval by consumers such as applications. It is intended to provide a reporting mechanism that is independent of other libraries such as ethdev.

- **Added bit-rate calculation library.**

  Added a library that can be used to calculate device bit-rates. Calculated bitrates are reported using the metrics library.

- **Added latency stats library.**

  Added a library that measures packet latency. The collected statistics are jitter and latency. For latency the minimum, average, and maximum is measured.

- **Added NXP DPAA2 SEC crypto PMD.**

  A new "dpaa2_sec" hardware based crypto PMD for NXP DPAA2 devices has been added. See the "Crypto Device Drivers" document for more details on this driver.

- **Updated the Cryptodev Scheduler PMD.**

  - Added a packet-size based distribution mode, which distributes the enqueued crypto operations among two slaves, based on their data lengths.
  - Added fail-over scheduling mode, which enqueues crypto operations to a primary slave first. Then, any operation that cannot be enqueued is enqueued to a secondary slave.
  - Added mode specific option support, so each scheduling mode can now be configured individually by the new API.

- **Updated the QAT PMD.**

  The QAT PMD has been updated with additional support for:

  - AES DOCSIS BPI algorithm.
  - DES DOCSIS BPI algorithm.
  - ZUC EEA3/EIA3 algorithms.

- **Updated the AESNI MB PMD.**

  The AESNI MB PMD has been updated with additional support for:

  - AES DOCSIS BPI algorithm.

- **Updated the OpenSSL PMD.**

  The OpenSSL PMD has been updated with additional support for:

  - DES DOCSIS BPI algorithm.

7.2 Resolved Issues

- **l2fwd-keepalive: Fixed unclean shutdowns.**

  Added clean shutdown to l2fwd-keepalive so that it can free up stale resources used for inter-process communication.

7.3 Known Issues

- **LSC interrupt doesn't work for virtio-user + vhost-kernel.**

  LSC interrupt cannot be detected when setting the backend, tap device, up/down as we fail to find a way to monitor such event.

7.4 API Changes

- The LPM `next_hop` field is extended from 8 bits to 21 bits for IPv6 while keeping ABI compatibility.
- **Reworked rte_ring library.**

  The rte_ring library has been reworked and updated. The following changes have been made to it:

  - Removed the build-time setting `CONFIG_RTE_RING_SPLIT_PROD_CONS`.
  - Removed the build-time setting `CONFIG_RTE_LIBRTE_RING_DEBUG`.
  - Removed the build-time setting `CONFIG_RTE_RING_PAUSE_REP_COUNT`.
  - Removed the function `rte_ring_set_water_mark` as part of a general removal of watermarks support in the library.
  - Added an extra parameter to the burst/bulk enqueue functions to return the number of free spaces in the ring after enqueue. This can be used by an application to implement its own watermark functionality.
  - Added an extra parameter to the burst/bulk dequeue functions to return the number elements remaining in the ring after dequeue.
  - Changed the return value of the enqueue and dequeue bulk functions to match that of the burst equivalents. In all cases, ring functions which operate on multiple packets now return the number of elements enqueued or dequeued, as appropriate. The updated functions are:

    * `rte_ring_mp_enqueue_bulk`
    * `rte_ring_sp_enqueue_bulk`
    * `rte_ring_enqueue_bulk`
    * `rte_ring_mc_dequeue_bulk`
    * `rte_ring_sc_dequeue_bulk`
    * `rte_ring_dequeue_bulk`

    NOTE: the above functions all have different parameters as well as different return values, due to the other listed changes above. This means that all instances of the functions in existing code will be flagged by the compiler. The return value usage should be checked while fixing the compiler error due to the extra parameter.

- **Reworked rte_vhost library.**

The rte_vhost library has been reworked to make it generic enough so that the user could build other vhost-user drivers on top of it. To achieve this the following changes have been made:

- The following vhost-pmd APIs are removed:
  - `rte_eth_vhost_feature_disable`
  - `rte_eth_vhost_feature_enable`
  - `rte_eth_vhost_feature_get`
- The vhost API `rte_vhost_driver_callback_register(ops)` is reworked to be per vhost-user socket file. Thus, it takes one more argument: `rte_vhost_driver_callback_register(path, ops)`.
- The vhost API `rte_vhost_get_queue_num` is deprecated, instead, `rte_vhost_get_vring_num` should be used.
- The following macros are removed in `rte_virtio_net.h`
  - `VIRTIO_RXQ`
  - `VIRTIO_TXQ`
  - `VIRTIO_QNUM`
- The following net specific header files are removed in `rte_virtio_net.h`
  - `linux/virtio_net.h`
  - `sys/socket.h`
  - `linux/if.h`
  - `rte_ether.h`
- The vhost struct `virtio_net_device_ops` is renamed to `vhost_device_ops`
- The vhost API `rte_vhost_driver_session_start` is removed. Instead, `rte_vhost_driver_start` should be used, and there is no need to create a thread to call it.
- The vhost public header file `rte_virtio_net.h` is renamed to `rte_vhost.h`

7.5 ABI Changes

- **Reorganized the mbuf structure.**

  The order and size of the fields in the `mbuf` structure changed, as described in the *New Features* section.

- The `rte_cryptodev_info.sym` structure has a new field `max_nb_sessions_per_qp` to support drivers which may support a limited number of sessions per queue_pair.

7.6 Removed Items

- KNI vhost support has been removed.

- The dpdk_qat sample application has been removed.

7.7 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
+  librte_bitratestats.so.1
   librte_cfgfile.so.2
   librte_cmdline.so.2
   librte_cryptodev.so.2
   librte_distributor.so.1
+  librte_eal.so.4
   librte_ethdev.so.6
+  librte_eventdev.so.1
   librte_hash.so.2
   librte_ip_frag.so.1
   librte_jobstats.so.1
   librte_kni.so.2
   librte_kvargs.so.1
+  librte_latencystats.so.1
   librte_lpm.so.2
+  librte_mbuf.so.3
   librte_mempool.so.2
   librte_meter.so.1
+  librte_metrics.so.1
   librte_net.so.1
   librte_pdump.so.1
   librte_pipeline.so.3
   librte_pmd_bond.so.1
   librte_pmd_ring.so.2
   librte_port.so.3
   librte_power.so.1
   librte_reorder.so.1
   librte_ring.so.1
   librte_sched.so.1
   librte_table.so.2
   librte_timer.so.1
   librte_vhost.so.3
```

7.8 Tested Platforms

- Intel(R) platforms with Intel(R) NICs combinations
  - CPU
    - Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
    - Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
    - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
    - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
    - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz

    - Intel(R) Xeon(R) CPU E5-2658 v3 @ 2.20GHz
  - OS:
    - CentOS 7.2
    - Fedora 25
    - FreeBSD 11
    - Red Hat Enterprise Linux Server release 7.3
    - SUSE Enterprise Linux 12
    - Wind River Linux 8
    - Ubuntu 16.04
    - Ubuntu 16.10
  - NICs:
    - Intel(R) 82599ES 10 Gigabit Ethernet Controller
      - Firmware version: 0x61bf0001
      - Device id (pf/vf): 8086:10fb / 8086:10ed
      - Driver version: 4.0.1-k (ixgbe)
    - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
      - Firmware version: 0x800001cf
      - Device id (pf/vf): 8086:15ad / 8086:15a8
      - Driver version: 4.2.5 (ixgbe)
    - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
      - Firmware version: 5.05
      - Device id (pf/vf): 8086:1572 / 8086:154c
      - Driver version: 1.5.23 (i40e)
    - Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
      - Firmware version: 5.05
      - Device id (pf/vf): 8086:1572 / 8086:154c
      - Driver version: 1.5.23 (i40e)
    - Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
      - Firmware version: 5.05
      - Device id (pf/vf): 8086:1584 / 8086:154c
      - Driver version: 1.5.23 (i40e)
    - Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
      - Firmware version: 5.05
      - Device id (pf/vf): 8086:1583 / 8086:154c

            - Driver version: 1.5.23 (i40e)
        - Intel(R) Corporation I350 Gigabit Network Connection
            - Firmware version: 1.48, 0x800006e7
            - Device id (pf/vf): 8086:1521 / 8086:1520
            - Driver version: 5.2.13-k (igb)
- Intel(R) platforms with Mellanox(R) NICs combinations
    - Platform details:
        - Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
        - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
        - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
        - Intel(R) Xeon(R) CPU E5-2640 @ 2.50GHz
    - OS:
        - Red Hat Enterprise Linux Server release 7.3 (Maipo)
        - Red Hat Enterprise Linux Server release 7.2 (Maipo)
        - Ubuntu 16.10
        - Ubuntu 16.04
        - Ubuntu 14.04
    - MLNX_OFED: 4.0-2.0.0.0
    - NICs:
        - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1007
            - Firmware version: 2.40.5030
        - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013
            - Firmware version: 12.18.2000
        - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013
            - Firmware version: 12.18.2000
        - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013

    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)

            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.2000
        * Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.2000
        * Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.19.1200
        * Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.19.1200
- IBM(R) Power8(R) with Mellanox(R) NICs combinations
    - Platform details:
        * Processor: POWER8E (raw), AltiVec supported
        * type-model: 8247-22L
        * Firmware FW810.21 (SV810_108)
    - OS: Ubuntu 16.04 LTS PPC le
    - MLNX_OFED: 4.0-2.0.0.0
    - NICs:
        * Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013
            - Firmware version: 12.18.2000
        * Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013
            - Firmware version: 12.18.2000
        * Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1013

    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000
  * Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.2000

CHAPTER

EIGHT

DPDK RELEASE 17.02

8.1 New Features

- **Added support for representing buses in EAL**

  The `rte_bus` structure was introduced into the EAL. This allows for devices to be represented by buses they are connected to. A new bus can be added to DPDK by extending the `rte_bus` structure and implementing the scan and probe functions. Once a new bus is registered using the provided APIs, new devices can be detected and initialized using bus scan and probe callbacks.

  With this change, devices other than PCI or VDEV type can be represented in the DPDK framework.

- **Added generic EAL API for I/O device memory read/write operations.**

  This API introduces 8 bit, 16 bit, 32 bit and 64 bit I/O device memory read/write operations along with "relaxed" versions.

  Weakly-ordered architectures like ARM need an additional I/O barrier for device memory read/write access over PCI bus. By introducing the EAL abstraction for I/O device memory read/write access, the drivers can access I/O device memory in an architecture-agnostic manner. The relaxed version does not have an additional I/O memory barrier, which is useful in accessing the device registers of integrated controllers which is implicitly strongly ordered with respect to memory access.

- **Added generic flow API (rte_flow).**

  This API provides a generic means to configure hardware to match specific ingress or egress traffic, alter its behavior and query related counters according to any number of user-defined rules.

  In order to expose a single interface with an unambiguous behavior that is common to all poll-mode drivers (PMDs) the `rte_flow` API is slightly higher-level than the legacy filtering framework, which it encompasses and supersedes (including all functions and filter types) .

  See the Generic flow API documentation for more information.

- **Added firmware version get API.**

  Added a new function `rte_eth_dev_fw_version_get()` to fetch the firmware version for a given device.

- **Added APIs for MACsec offload support to the ixgbe PMD.**

Six new APIs have been added to the ixgbe PMD for MACsec offload support. The declarations for the APIs can be found in `rte_pmd_ixgbe.h`.

- **Added I219 NICs support.**

  Added support for I219 Intel 1GbE NICs.

- **Added VF Daemon (VFD) for i40e. - EXPERIMENTAL**

  This is an EXPERIMENTAL feature to enhance the capability of the DPDK PF as many VF management features are not currently supported by the kernel PF driver. Some new private APIs are implemented directly in the PMD without an abstraction layer. They can be used directly by some users who have the need.

  The new APIs to control VFs directly from PF include:

  – Set VF MAC anti-spoofing.

  – Set VF VLAN anti-spoofing.

  – Set TX loopback.

  – Set VF unicast promiscuous mode.

  – Set VF multicast promiscuous mode.

  – Set VF MTU.

  – Get/reset VF stats.

  – Set VF MAC address.

  – Set VF VLAN stripping.

  – Vf VLAN insertion.

  – Set VF broadcast mode.

  – Set VF VLAN tag.

  – Set VF VLAN filter.

  VFD also includes VF to PF mailbox message management from an application. When the PF receives mailbox messages from the VF the PF should call the callback provided by the application to know if they're permitted to be processed.

  As an EXPERIMENTAL feature, please be aware it can be changed or even removed without prior notice.

- **Updated the i40e base driver.**

  Updated the i40e base driver, including the following changes:

  – Replace existing legacy `memcpy()` calls with `i40e_memcpy()` calls.

  – Use `BIT()` macro instead of bit fields.

  – Add clear all WoL filters implementation.

  – Add broadcast promiscuous control per VLAN.

  – Remove unused `X722_SUPPORT` and `I40E_NDIS_SUPPORT` macros.

- **Updated the enic driver.**

  – Set new Rx checksum flags in mbufs to indicate unknown, good or bad checksums.

  - Fix set/remove of MAC addresses. Allow up to 64 addresses per device.
  - Enable TSO on outer headers.

- **Added Solarflare libefx-based network PMD.**

  Added a new network PMD which supports Solarflare SFN7xxx and SFN8xxx family of 10/40 Gbps adapters.

- **Updated the mlx4 driver.**
  - Addressed a few bugs.

- **Added support for Mellanox ConnectX-5 adapters (mlx5).**

  Added support for Mellanox ConnectX-5 family of 10/25/40/50/100 Gbps adapters to the existing mlx5 PMD.

- **Updated the mlx5 driver.**
  - Improve Tx performance by using vector logic.
  - Improve RSS balancing when number of queues is not a power of two.
  - Generic flow API support for Ethernet, IPv4, IPv4, UDP, TCP, VLAN and VXLAN pattern items with DROP and QUEUE actions.
  - Support for extended statistics.
  - Addressed several data path bugs.
  - As of MLNX_OFED 4.0-1.0.1.0, the Toeplitz RSS hash function is not symmetric anymore for consistency with other PMDs.

- **virtio-user with vhost-kernel as another exceptional path.**

  Previously, we upstreamed a virtual device, virtio-user with vhost-user as the backend as a way of enabling IPC (Inter-Process Communication) and user space container networking.

  Virtio-user with vhost-kernel as the backend is a solution for the exception path, such as KNI, which exchanges packets with the kernel networking stack. This solution is very promising in:

  - Maintenance: vhost and vhost-net (kernel) is an upstreamed and extensively used kernel module.
  - Features: vhost-net is designed to be a networking solution, which has lots of networking related features, like multi-queue, TSO, multi-seg mbuf, etc.
  - Performance: similar to KNI, this solution would use one or more kthreads to send/receive packets from user space DPDK applications, which has little impact on user space polling thread (except that it might enter into kernel space to wake up those kthreads if necessary).

- **Added virtio Rx interrupt support.**

  Added a feature to enable Rx interrupt mode for virtio pci net devices as bound to VFIO (noiommu mode) and driven by virtio PMD.

  With this feature, the virtio PMD can switch between polling mode and interrupt mode, to achieve best performance, and at the same time save power. It can work on both legacy

and modern virtio devices. In this mode, each `rxq` is mapped with an excluded MSIx interrupt.

See the Virtio Interrupt Mode documentation for more information.

- **Added ARMv8 crypto PMD.**

  A new crypto PMD has been added, which provides combined mode cryptographic operations optimized for ARMv8 processors. The driver can be used to enhance performance in processing chained operations such as cipher + HMAC.

- **Updated the QAT PMD.**

  The QAT PMD has been updated with additional support for:

  - DES algorithm.
  - Scatter-gather list (SGL) support.

- **Updated the AESNI MB PMD.**

  - The Intel(R) Multi Buffer Crypto for IPsec library used in AESNI MB PMD has been moved to a new repository, in GitHub.
  - Support has been added for single operations (cipher only and authentication only).

- **Updated the AES-NI GCM PMD.**

  The AES-NI GCM PMD was migrated from the Multi Buffer library to the ISA-L library. The migration entailed adding additional support for:

  - GMAC algorithm.
  - 256-bit cipher key.
  - Session-less mode.
  - Out-of place processing
  - Scatter-gather support for chained mbufs (only out-of place and destination mbuf must be contiguous)

- **Added crypto performance test application.**

  Added a new performance test application for measuring performance parameters of PMDs available in the crypto tree.

- **Added Elastic Flow Distributor library (rte_efd).**

  Added a new library which uses perfect hashing to determine a target/value for a given incoming flow key.

  The library does not store the key itself for lookup operations, and therefore, lookup performance is not dependent on the key size. Also, the target/value can be any arbitrary value (8 bits by default). Finally, the storage requirement is much smaller than a hash-based flow table and therefore, it can better fit in CPU cache and scale to millions of flow keys.

  See the Elastic Flow Distributor Library documentation in the Programmers Guide document, for more information.

8.2 Resolved Issues

8.2.1 Drivers

- **net/virtio: Fixed multiple process support.**

  Fixed a few regressions introduced in recent releases that break the virtio multiple process support.

8.2.2 Examples

- **examples/ethtool: Fixed crash with non-PCI devices.**

  Fixed issue where querying a non-PCI device was dereferencing non-existent PCI data resulting in a segmentation fault.

8.3 API Changes

- **Moved five APIs for VF management from the ethdev to the ixgbe PMD.**

  The following five APIs for VF management from the PF have been removed from the ethdev, renamed, and added to the ixgbe PMD:

  ```
  rte_eth_dev_set_vf_rate_limit()
  rte_eth_dev_set_vf_rx()
  rte_eth_dev_set_vf_rxmode()
  rte_eth_dev_set_vf_tx()
  rte_eth_dev_set_vf_vlan_filter()
  ```

  The API's have been renamed to the following:

  ```
  rte_pmd_ixgbe_set_vf_rate_limit()
  rte_pmd_ixgbe_set_vf_rx()
  rte_pmd_ixgbe_set_vf_rxmode()
  rte_pmd_ixgbe_set_vf_tx()
  rte_pmd_ixgbe_set_vf_vlan_filter()
  ```

  The declarations for the API's can be found in `rte_pmd_ixgbe.h`.

8.4 ABI Changes

8.5 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
   librte_acl.so.2
   librte_cfgfile.so.2
   librte_cmdline.so.2
   librte_cryptodev.so.2
   librte_distributor.so.1
   librte_eal.so.3
 + librte_ethdev.so.6
   librte_hash.so.2
   librte_ip_frag.so.1
```

```
librte_jobstats.so.1
librte_kni.so.2
librte_kvargs.so.1
librte_lpm.so.2
librte_mbuf.so.2
librte_mempool.so.2
librte_meter.so.1
librte_net.so.1
librte_pdump.so.1
librte_pipeline.so.3
librte_pmd_bond.so.1
librte_pmd_ring.so.2
librte_port.so.3
librte_power.so.1
librte_reorder.so.1
librte_ring.so.1
librte_sched.so.1
librte_table.so.2
librte_timer.so.1
librte_vhost.so.3
```

8.6 Tested Platforms

This release has been tested with the below list of CPU/device/firmware/OS. Each section describes a different set of combinations.

- Intel(R) platforms with Mellanox(R) NICs combinations
  - Platform details
    - Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz
    - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
  - OS:
    - CentOS 7.0
    - Fedora 23
    - Fedora 24
    - FreeBSD 10.3
    - Red Hat Enterprise Linux 7.2
    - SUSE Enterprise Linux 12
    - Ubuntu 14.04 LTS
    - Ubuntu 15.10
    - Ubuntu 16.04 LTS
    - Wind River Linux 8
  - MLNX_OFED: 4.0-1.0.1.0
  - NICs:
    - Mellanox(R) ConnectX(R)-3 Pro 40G MCX354A-FCC_Ax (2x40G)

      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1007
      - Firmware version: 2.40.5030
    * Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
      - Host interface: PCI Express 3.0 x16
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013
      - Firmware version: 12.18.1000
    * Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
      - Host interface: PCI Express 3.0 x8
      - Device ID: 15b3:1013

            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.1000
        - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.1000
        - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.18.1000
        - Mellanox(R) ConnectX-5 Ex EN 100G MCX516A-CDAT (2x100G)
            - Host interface: PCI Express 4.0 x16
            - Device ID: 15b3:1019
            - Firmware version: 16.18.1000
- IBM(R) Power8(R) with Mellanox(R) NICs combinations
    - Machine:
        - Processor: POWER8E (raw), AltiVec supported
            - type-model: 8247-22L
            - Firmware FW810.21 (SV810_108)

- OS: Ubuntu 16.04 LTS PPC le
- MLNX_OFED: 4.0-1.0.1.0
- NICs:
  - Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - Firmware version: 12.18.1000
  - Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013

            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1013
            - Firmware version: 12.18.1000
        - Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.1000
        - Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
            - Host interface: PCI Express 3.0 x8
            - Device ID: 15b3:1015
            - Firmware version: 14.18.1000
        - Mellanox(R) ConnectX(R)-5 100G MCX556A-ECAT (2x100G)
            - Host interface: PCI Express 3.0 x16
            - Device ID: 15b3:1017
            - Firmware version: 16.18.1000
- Intel(R) platforms with Intel(R) NICs combinations
    - Platform details
        - Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
        - Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
        - Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
        - Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
        - Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
        - Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
        - Intel(R) Xeon(R) CPU E5-2658 v2 @ 2.40GHz

- OS:
  - CentOS 7.2
  - Fedora 25
  - FreeBSD 11
  - Red Hat Enterprise Linux Server release 7.3
  - SUSE Enterprise Linux 12
  - Wind River Linux 8
  - Ubuntu 16.04
  - Ubuntu 16.10
- NICs:
  - Intel(R) 82599ES 10 Gigabit Ethernet Controller
    - Firmware version: 0x61bf0001
    - Device id (pf/vf): 8086:10fb / 8086:10ed
    - Driver version: 4.0.1-k (ixgbe)
  - Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
    - Firmware version: 0x800001cf
    - Device id (pf/vf): 8086:15ad / 8086:15a8
    - Driver version: 4.2.5 (ixgbe)
  - Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
    - Firmware version: 5.05
    - Device id (pf/vf): 8086:1572 / 8086:154c
    - Driver version: 1.5.23 (i40e)
  - Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
    - Firmware version: 5.05
    - Device id (pf/vf): 8086:1572 / 8086:154c
    - Driver version: 1.5.23 (i40e)
  - Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
    - Firmware version: 5.05
    - Device id (pf/vf): 8086:1584 / 8086:154c
    - Driver version: 1.5.23 (i40e)
  - Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
    - Firmware version: 5.05
    - Device id (pf/vf): 8086:1583 / 8086:154c
    - Driver version: 1.5.23 (i40e)

* Intel(R) Corporation I350 Gigabit Network Connection
  - Firmware version: 1.48, 0x800006e7
  - Device id (pf/vf): 8086:1521 / 8086:1520
  - Driver version: 5.2.13-k (igb)

CHAPTER

NINE

DPDK RELEASE 16.11

9.1 New Features

- **Added software parser for packet type.**
  - Added a new function `rte_pktmbuf_read()` to read the packet data from an mbuf chain, linearizing if required.
  - Added a new function `rte_net_get_ptype()` to parse an Ethernet packet in an mbuf chain and retrieve its packet type from software.
  - Added new functions `rte_get_ptype_*()` to dump a packet type as a string.
- **Improved offloads support in mbuf.**
  - Added a new function `rte_raw_cksum_mbuf()` to process the checksum of data embedded in an mbuf chain.
  - Added new Rx checksum flags in mbufs to describe more states: unknown, good, bad, or not present (useful for virtual drivers). This modification was done for IP and L4.
  - Added a new Rx LRO mbuf flag, used when packets are coalesced. This flag indicates that the segment size of original packets is known.
- **Added vhost-user dequeue zero copy support.**

  The copy in the dequeue path is avoided in order to improve the performance. In the VM2VM case, the boost is quite impressive. The bigger the packet size, the bigger performance boost you may get. However, for the VM2NIC case, there are some limitations, so the boost is not as impressive as the VM2VM case. It may even drop quite a bit for small packets.

  For that reason, this feature is disabled by default. It can be enabled when the `RTE_VHOST_USER_DEQUEUE_ZERO_COPY` flag is set. Check the VHost section of the Programming Guide for more information.
- **Added vhost-user indirect descriptors support.**

  If the indirect descriptor feature is enabled, each packet sent by the guest will take exactly one slot in the enqueue virtqueue. Without this feature, as in the current version, even 64 bytes packets take two slots with Virtio PMD on guest side.

  The main impact is better performance for 0% packet loss use-cases, as it behaves as if the virtqueue size was enlarged, so more packets can be buffered in the case of system

perturbations. On the downside, small performance degradations were measured when running micro-benchmarks.

- **Added vhost PMD xstats.**

  Added extended statistics to vhost PMD from a per port perspective.

- **Supported offloads with virtio.**

  Added support for the following offloads in virtio:

  - Rx/Tx checksums.
  - LRO.
  - TSO.

- **Added virtio NEON support for ARM.**

  Added NEON support for ARM based virtio.

- **Updated the ixgbe base driver.**

  Updated the ixgbe base driver, including the following changes:

  - Added X550em_a 10G PHY support.
  - Added support for flow control auto negotiation for X550em_a 1G PHY.
  - Added X550em_a FW ALEF support.
  - Increased mailbox version to `ixgbe_mbox_api_13`.
  - Added two MAC operations for Hyper-V support.

- **Added APIs for VF management to the ixgbe PMD.**

  Eight new APIs have been added to the ixgbe PMD for VF management from the PF. The declarations for the API's can be found in `rte_pmd_ixgbe.h`.

- **Updated the enic driver.**

  - Added update to use interrupt for link status checking instead of polling.
  - Added more flow director modes on UCS Blade with firmware version >= 2.0(13e).
  - Added full support for MTU update.
  - Added support for the `rte_eth_rx_queue_count` function.

- **Updated the mlx5 driver.**

  - Added support for RSS hash results.
  - Added several performance improvements.
  - Added several bug fixes.

- **Updated the QAT PMD.**

  The QAT PMD was updated with additional support for:

  - MD5_HMAC algorithm.
  - SHA224-HMAC algorithm.
  - SHA384-HMAC algorithm.

  - GMAC algorithm.
  - KASUMI (F8 and F9) algorithm.
  - 3DES algorithm.
  - NULL algorithm.
  - C3XXX device.
  - C62XX device.

- **Added openssl PMD.**

  A new crypto PMD has been added, which provides several ciphering and hashing algorithms. All cryptography operations use the Openssl library crypto API.

- **Updated the IPsec example.**

  Updated the IPsec example with the following support:

  - Configuration file support.
  - AES CBC IV generation with cipher forward function.
  - AES GCM/CTR mode.

- **Added support for new gcc -march option.**

  The GCC 4.9 `-march` option supports the Intel processor code names. The config option `RTE_MACHINE` can be used to pass code names to the compiler via the `-march` flag.

9.2 Resolved Issues

9.2.1 Drivers

- **enic: Fixed several flow director issues.**
- **enic: Fixed inadvertent setting of L4 checksum ptype on ICMP packets.**
- **enic: Fixed high driver overhead when servicing Rx queues beyond the first.**

9.3 Known Issues

- **L3fwd-power app does not work properly when Rx vector is enabled.**

  The L3fwd-power app doesn't work properly with some drivers in vector mode since the queue monitoring works differently between scalar and vector modes leading to incorrect frequency scaling. In addition, L3fwd-power application requires the mbuf to have correct packet type set but in some drivers the vector mode must be disabled for this.

  Therefore, in order to use L3fwd-power, vector mode should be disabled via the config file.

- **Digest address must be supplied for crypto auth operation on QAT PMD.**

  The cryptodev API specifies that if the rte_crypto_sym_op.digest.data field, and by inference the digest.phys_addr field which points to the same location, is not set for an auth

operation the driver is to understand that the digest result is located immediately following the region over which the digest is computed. The QAT PMD doesn't correctly handle this case and reads and writes to an incorrect location.

Callers can workaround this by always supplying the digest virtual and physical address fields in the rte_crypto_sym_op for an auth operation.

9.4 API Changes

- The driver naming convention has been changed to make them more consistent. It especially impacts `--vdev` arguments. For example `eth_pcap` becomes `net_pcap` and `cryptodev_aesni_mb_pmd` becomes `crypto_aesni_mb`.

  For backward compatibility an alias feature has been enabled to support the original names.

- The log history has been removed.

- The `rte_ivshmem` feature (including library and EAL code) has been removed in 16.11 because it had some design issues which were not planned to be fixed.

- The `file_name` data type of `struct rte_port_source_params` and `struct rte_port_sink_params` is changed from `char *` to `const char *`.

- **Improved device/driver hierarchy and generalized hotplugging.**

  The device and driver relationship has been restructured by introducing generic classes. This paves the way for having PCI, VDEV and other device types as instantiated objects rather than classes in themselves. Hotplugging has also been generalized into EAL so that Ethernet or crypto devices can use the common infrastructure.

  - Removed `pmd_type` as a way of segregation of devices.
  - Moved `numa_node` and `devargs` into `rte_driver` from `rte_pci_driver`. These can now be used by any instantiated object of `rte_driver`.
  - Added `rte_device` class and all PCI and VDEV devices inherit from it
  - Renamed devinit/devuninit handlers to probe/remove to make it more semantically correct with respect to the device <=> driver relationship.
  - Moved hotplugging support to EAL. Hereafter, PCI and vdev can use the APIs `rte_eal_dev_attach` and `rte_eal_dev_detach`.
  - Renamed helpers and support macros to make them more synonymous with their device types (e.g. `PMD_REGISTER_DRIVER` => `RTE_PMD_REGISTER_PCI`).
  - Device naming functions have been generalized from ethdev and cryptodev to EAL. `rte_eal_pci_device_name` has been introduced for obtaining unique device name from PCI Domain-BDF description.
  - Virtual device registration APIs have been added: `rte_eal_vdrv_register` and `rte_eal_vdrv_unregister`.

9.5 ABI Changes

9.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
    librte_acl.so.2
    librte_cfgfile.so.2
    librte_cmdline.so.2
+   librte_cryptodev.so.2
    librte_distributor.so.1
+   librte_eal.so.3
+   librte_ethdev.so.5
    librte_hash.so.2
    librte_ip_frag.so.1
    librte_jobstats.so.1
    librte_kni.so.2
    librte_kvargs.so.1
    librte_lpm.so.2
    librte_mbuf.so.2
    librte_mempool.so.2
    librte_meter.so.1
    librte_net.so.1
    librte_pdump.so.1
    librte_pipeline.so.3
    librte_pmd_bond.so.1
    librte_pmd_ring.so.2
    librte_port.so.3
    librte_power.so.1
    librte_reorder.so.1
    librte_ring.so.1
    librte_sched.so.1
    librte_table.so.2
    librte_timer.so.1
    librte_vhost.so.3
```

9.7 Tested Platforms

1. SuperMicro 1U
   - BIOS: 1.0c
   - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
   - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
     - Firmware-version: 0x800001cf
     - Device ID (PF/VF): 8086:15ad /8086:15a8
   - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 2U

   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
4. Intel(R) Server board S2600GZ
   - BIOS: SE5C600.86B.02.02.0002.122320131210
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR
   - BIOS: SE5C600.86B.02.01.0002.082220131453
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
6. Intel(R) Server board S2600CWT
   - BIOS: SE5C610.86B.01.01.0009.060120151350
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.01.01.0005.101720141054
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.11.01.0044.090120151156
   - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
9. Intel(R) Server board S2600WTT
   - Processor: Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz
10. Intel(R) Server
    - Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz
11. IBM(R) Power8(R)
    - Machine type-model: 8247-22L
    - Firmware FW810.21 (SV810_108)
    - Processor: POWER8E (raw), AltiVec supported

9.8 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
   - Firmware version: 0x80000389
   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)
2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed

   - Driver version: 4.0.1-k (ixgbe)

3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.5.23 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.5.23 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1584 / 8086:154c
   - Driver version: 1.5.23 (i40e)
7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
   - Firmware version: 5.05
   - Device id (pf/vf): 8086:1583 / 8086:154c
   - Driver version: 1.5.23 (i40e)
8. Intel(R) Corporation I350 Gigabit Network Connection
   - Firmware version: 1.48, 0x800006e7
   - Device id (pf/vf): 8086:1521 / 8086:1520
   - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Multi-host Controller FM10000
   - Firmware version: N/A
   - Device id (pf/vf): 8086:15d0
   - Driver version: 0.17.0.9 (fm10k)
10. Mellanox(R) ConnectX(R)-4 10G MCX4111A-XCAT (1x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010

11. Mellanox(R) ConnectX(R)-4 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
12. Mellanox(R) ConnectX(R)-4 25G MCX4111A-ACAT (1x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
13. Mellanox(R) ConnectX(R)-4 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
14. Mellanox(R) ConnectX(R)-4 40G MCX4131A-BCAT/MCX413A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
15. Mellanox(R) ConnectX(R)-4 40G MCX415A-BCAT (1x40G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
16. Mellanox(R) ConnectX(R)-4 50G MCX4131A-GCAT/MCX413A-GCAT (1x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
17. Mellanox(R) ConnectX(R)-4 50G MCX414A-BCAT (2x50G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0

- Firmware version: 12.17.1010

18. Mellanox(R) ConnectX(R)-4 50G MCX415A-GCAT/MCX416A-BCAT/MCX416A-GCAT (2x50G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
19. Mellanox(R) ConnectX(R)-4 50G MCX415A-CCAT (1x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
20. Mellanox(R) ConnectX(R)-4 100G MCX416A-CCAT (2x100G)
    - Host interface: PCI Express 3.0 x16
    - Device ID: 15b3:1013
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 12.17.1010
21. Mellanox(R) ConnectX(R)-4 Lx 10G MCX4121A-XCAT (2x10G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1015
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 14.17.1010
22. Mellanox(R) ConnectX(R)-4 Lx 25G MCX4121A-ACAT (2x25G)
    - Host interface: PCI Express 3.0 x8
    - Device ID: 15b3:1015
    - MLNX_OFED: 3.4-1.0.0.0
    - Firmware version: 14.17.1010

9.9 Tested OSes

- CentOS 7.2
- Fedora 23
- Fedora 24
- FreeBSD 10.3
- FreeBSD 11

- Red Hat Enterprise Linux Server release 6.7 (Santiago)
- Red Hat Enterprise Linux Server release 7.0 (Maipo)
- Red Hat Enterprise Linux Server release 7.2 (Maipo)
- SUSE Enterprise Linux 12
- Wind River Linux 6.0.0.26
- Wind River Linux 8
- Ubuntu 14.04
- Ubuntu 15.04
- Ubuntu 16.04

CHAPTER

TEN

DPDK RELEASE 16.07

10.1 New Features

- **Removed the mempool cache memory if caching is not being used.**

  The size of the mempool structure is reduced if the per-lcore cache is disabled.

- **Added mempool external cache for non-EAL thread.**

  Added new functions to create, free or flush a user-owned mempool cache for non-EAL threads. Previously the caching was always disabled on these threads.

- **Changed the memory allocation scheme in the mempool library.**

  - Added the ability to allocate a large mempool in fragmented virtual memory.
  - Added new APIs to populate a mempool with memory.
  - Added an API to free a mempool.
  - Modified the API of the `rte_mempool_obj_iter()` function.
  - Dropped the specific Xen Dom0 code.
  - Dropped the specific anonymous mempool code in testpmd.

- **Added a new driver for Broadcom NetXtreme-C devices.**

  Added the new bnxt driver for Broadcom NetXtreme-C devices. See the "Network Interface Controller Drivers" document for more details on this new driver.

- **Added a new driver for ThunderX nicvf devices.**

  Added the new thunderx net driver for ThunderX nicvf devices. See the "Network Interface Controller Drivers" document for more details on this new driver.

- **Added mailbox interrupt support for ixgbe and igb VFs.**

  When the physical NIC link comes up or down, the PF driver will send a mailbox message to notify each VF. To handle this link up/down event, support have been added for a mailbox interrupt to receive the message and allow the application to register a callback for it.

- **Updated the ixgbe base driver.**

  The ixgbe base driver was updated with changes including the following:

  - Added sgmii link for X550.
  - Added MAC link setup for X550a SFP and SFP+.

  - Added KR support for X550em_a.
  - Added new PHY definitions for M88E1500.
  - Added support for the VLVF to be bypassed when adding/removing a VFTA entry.
  - Added X550a flow control auto negotiation support.

- **Updated the i40e base driver.**

  Updated the i40e base driver including support for new devices IDs.

- **Updated the enic driver.**

  The enic driver was updated with changes including the following:

  - Optimized the Tx function.
  - Added Scattered Rx capability.
  - Improved packet type identification.
  - Added MTU update in non Scattered Rx mode and enabled MTU of up to 9208 with UCS Software release 2.2 on 1300 series VICs.

- **Updated the mlx5 driver.**

  The mlx5 driver was updated with changes including the following:

  - Data path was refactored to bypass Verbs to improve RX and TX performance.
  - Removed compilation parameters for inline send, `MLX5_MAX_INLINE`, and added command line parameter instead, `txq_inline`.
  - Improved TX scatter gather support: Removed compilation parameter `MLX5_PMD_SGE_WR_N`. Scatter-gather elements is set to the maximum value the NIC supports. Removed linearization logic, this decreases the memory consumption of the PMD.
  - Improved jumbo frames support, by dynamically setting RX scatter gather elements according to the MTU and mbuf size, no need for compilation parameter `MLX5_PMD_SGE_WR_N`

- **Added support for virtio on IBM POWER8.**

  The ioports are mapped in memory when using Linux UIO.

- **Added support for Virtio in containers.**

  Add a new virtual device, named virtio_user, to support virtio for containers.

  Known limitations:

  - Control queue and multi-queue are not supported yet.
  - Doesn't work with `--huge-unlink`.
  - Doesn't work with `--no-huge`.
  - Doesn't work when there are more than `VHOST_MEMORY_MAX_NREGIONS(8)` hugepages.
  - Root privilege is required for sorting hugepages by physical address.
  - Can only be used with the vhost user backend.

- **Added vhost-user client mode.**

  DPDK vhost-user now supports client mode as well as server mode. Client mode is enabled when the `RTE_VHOST_USER_CLIENT` flag is set while calling `rte_vhost_driver_register`.

  When DPDK vhost-user restarts from an normal or abnormal exit (such as a crash), the client mode allows DPDK to establish the connection again. Note that QEMU version v2.7 or above is required for this feature.

  DPDK vhost-user will also try to reconnect by default when:

  – The first connect fails (for example when QEMU is not started yet).

  – The connection is broken (for example when QEMU restarts).

  It can be turned off by setting the `RTE_VHOST_USER_NO_RECONNECT` flag.

- **Added NSH packet recognition in i40e.**

- **Added AES-CTR support to AESNI MB PMD.**

  Now AESNI MB PMD supports 128/192/256-bit counter mode AES encryption and decryption.

- **Added AES counter mode support for Intel QuickAssist devices.**

  Enabled support for the AES CTR algorithm for Intel QuickAssist devices. Provided support for algorithm-chaining operations.

- **Added KASUMI SW PMD.**

  A new Crypto PMD has been added, which provides KASUMI F8 (UEA1) ciphering and KASUMI F9 (UIA1) hashing.

- **Added multi-writer support for RTE Hash with Intel TSX.**

  The following features/modifications have been added to rte_hash library:

  – Enabled application developers to use an extra flag for `rte_hash` creation to specify default behavior (multi-thread safe/unsafe) with the `rte_hash_add_key` function.

  – Changed the Cuckoo Hash Search algorithm to breadth first search for multi-writer routines and split Cuckoo Hash Search and Move operations in order to reduce transactional code region and improve TSX performance.

  – Added a hash multi-writer test case to the test app.

- **Improved IP Pipeline Application.**

  The following features have been added to the ip_pipeline application:

  – Configure the MAC address in the routing pipeline and automatic route updates with change in link state.

  – Enable RSS per network interface through the configuration file.

  – Streamline the CLI code.

- **Added keepalive enhancements.**

  Added support for reporting of core states other than “dead” to monitoring applications, enabling the support of broader liveness reporting to external processes.

- **Added packet capture framework.**
  - A new library `librte_pdump` is added to provide a packet capture API.
  - A new `app/pdump` tool is added to demonstrate capture packets in DPDK.
- **Added floating VEB support for i40e PF driver.**

  A "floating VEB" is a special Virtual Ethernet Bridge (VEB) which does not have an upload port, but instead is used for switching traffic between virtual functions (VFs) on a port.

  For information on this feature, please see the "I40E Poll Mode Driver" section of the "Network Interface Controller Drivers" document.
- **Added support for live migration of a VM with SRIOV VF.**

  Live migration of a VM with Virtio and VF PMD's using the bonding PMD.

10.2 Resolved Issues

10.2.1 EAL

- **igb_uio: Fixed possible mmap failure for Linux >= 4.5.**

  The mmaping of the iomem range of the PCI device fails for kernels that enabled the `CONFIG_IO_STRICT_DEVMEM` option. The error seen by the user is as similar to the following:

  ```
  EAL: pci_map_resource():

      cannot mmap(39, 0x7f1c51800000, 0x100000, 0x0):
      Invalid argument (0xffffffffffffffff)
  ```

  The `CONFIG_IO_STRICT_DEVMEM` kernel option was introduced in Linux v4.5.

  The issues was resolve by updating `igb_uio` to stop reserving PCI memory resources. From the kernel point of view the iomem region looks like idle and mmap works again. This matches the `uio_pci_generic` usage.

10.2.2 Drivers

- **i40e: Fixed vlan stripping from inner header.**

  Previously, for tunnel packets, such as VXLAN/NVGRE, the vlan tags of the inner header will be stripped without putting vlan info to descriptor. Now this issue is fixed by disabling vlan stripping from inner header.
- **i40e: Fixed the type issue of a single VLAN type.**

  Currently, if a single VLAN header is added in a packet, it's treated as inner VLAN. But generally, a single VLAN header is treated as the outer VLAN header. This issue is fixed by changing corresponding register for single VLAN.
- **enic: Fixed several issues when stopping then restarting ports and queues.**

  Fixed several crashes related to stopping then restarting ports and queues. Fixed possible crash when re-configuring the number of Rx queue descriptors.

- **enic: Fixed Rx data mis-alignment if mbuf data offset modified.**

  Fixed possible Rx corruption when mbufs were returned to a pool with data offset other than RTE_PKTMBUF_HEADROOM.

- **enic: Fixed Tx IP/UDP/TCP checksum offload and VLAN insertion.**

- **enic: Fixed Rx error and missed counters.**

10.2.3 Libraries

- **mbuf: Fixed refcnt update when detaching.**

  Fix the `rte_pktmbuf_detach()` function to decrement the direct mbuf's reference counter. The previous behavior was not to affect the reference counter. This lead to a memory leak of the direct mbuf.

10.2.4 Examples

10.2.5 Other

10.3 Known Issues

10.4 API Changes

- The following counters are removed from the `rte_eth_stats` structure:
  - `ibadcrc`
  - `ibadlen`
  - `imcasts`
  - `fdirmatch`
  - `fdirmiss`
  - `tx_pause_xon`
  - `rx_pause_xon`
  - `tx_pause_xoff`
  - `rx_pause_xoff`
- The extended statistics are fetched by ids with `rte_eth_xstats_get` after a lookup by name `rte_eth_xstats_get_names`.
- The function `rte_eth_dev_info_get` fill the new fields `nb_rx_queues` and `nb_tx_queues` in the structure `rte_eth_dev_info`.
- The vhost function `rte_vring_available_entries` is renamed to `rte_vhost_avail_entries`.
- All existing vhost APIs and callbacks with `virtio_net` struct pointer as the parameter have been changed due to the ABI refactoring described below. It is replaced by `int vid`.

- The function `rte_vhost_enqueue_burst` no longer supports concurrent enqueuing packets to the same queue.
- The function `rte_eth_dev_set_mtu` adds a new return value `-EBUSY`, which indicates the operation is forbidden because the port is running.
- The script `dpdk_nic_bind.py` is renamed to `dpdk-devbind.py`. And the script `setup.sh` is renamed to `dpdk-setup.sh`.

10.5 ABI Changes

- The `rte_port_source_params` structure has new fields to support PCAP files. It was already in release 16.04 with `RTE_NEXT_ABI` flag.
- The `rte_eth_dev_info` structure has new fields `nb_rx_queues` and `nb_tx_queues` to support the number of queues configured by software.
- A Vhost ABI refactoring has been made: the `virtio_net` structure is no longer exported directly to the application. Instead, a handle, `vid`, has been used to represent this structure internally.

10.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.4
  librte_acl.so.2
  librte_cfgfile.so.2
  librte_cmdline.so.2
  librte_cryptodev.so.1
  librte_distributor.so.1
  librte_eal.so.2
  librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
  librte_kni.so.2
  librte_kvargs.so.1
  librte_lpm.so.2
  librte_mbuf.so.2
+ librte_mempool.so.2
  librte_meter.so.1
  librte_pdump.so.1
  librte_pipeline.so.3
  librte_pmd_bond.so.1
  librte_pmd_ring.so.2
+ librte_port.so.3
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
  librte_table.so.2
  librte_timer.so.1
+ librte_vhost.so.3
```

10.7 Tested Platforms

1. SuperMicro 1U
    - BIOS: 1.0c
    - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
    - BIOS: 1.0a
    - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
    - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
        - Firmware-version: 0x800001cf
        - Device ID (PF/VF): 8086:15ad /8086:15a8
    - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 2U
    - BIOS: 1.0a
    - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
4. Intel(R) Server board S2600GZ
    - BIOS: SE5C600.86B.02.02.0002.122320131210
    - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR
    - BIOS: SE5C600.86B.02.01.0002.082220131453
    - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
6. Intel(R) Server board S2600CWT
    - BIOS: SE5C610.86B.01.01.0009.060120151350
    - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
    - BIOS: SE5C610.86B.01.01.0005.101720141054
    - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
    - BIOS: SE5C610.86B.11.01.0044.090120151156
    - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz

10.8 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
    - Firmware version: 0x80000389

   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)
2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed
   - Driver version: 4.0.1-k (ixgbe)
3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.26 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.25 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1584 / 8086:154c
   - Driver version: 1.4.25 (i40e)
7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
   - Firmware version: 5.04
   - Device id (pf/vf): 8086:1583 / 8086:154c
   - Driver version: 1.4.25 (i40e)
8. Intel(R) Corporation I350 Gigabit Network Connection
   - Firmware version: 1.48, 0x800006e7
   - Device id (pf/vf): 8086:1521 / 8086:1520
   - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Multi-host Controller FM10000
   - Firmware version: N/A
   - Device id (pf/vf): 8086:15d0
   - Driver version: 0.17.0.9 (fm10k)

10.9 Tested OSes

- CentOS 7.0
- Fedora 23
- Fedora 24
- FreeBSD 10.3
- Red Hat Enterprise Linux 7.2
- SUSE Enterprise Linux 12
- Ubuntu 15.10
- Ubuntu 16.04 LTS
- Wind River Linux 8

CHAPTER

ELEVEN

DPDK RELEASE 16.04

11.1 New Features

- **Added function to check primary process state.**

  A new function `rte_eal_primary_proc_alive()` has been added to allow the user to detect if a primary process is running. Use cases for this feature include fault detection, and monitoring using secondary processes.

- **Enabled bulk allocation of mbufs.**

  A new function `rte_pktmbuf_alloc_bulk()` has been added to allow the user to bulk allocate mbufs.

- **Added device link speed capabilities.**

  The structure `rte_eth_dev_info` now has a `speed_capa` bitmap, which allows the application to determine the supported speeds of each device.

- **Added bitmap of link speeds to advertise.**

  Added a feature to allow the definition of a set of advertised speeds for auto-negotiation, explicitly disabling link auto-negotiation (single speed) and full auto-negotiation.

- **Added new poll-mode driver for Amazon Elastic Network Adapters (ENA).**

  The driver operates for a variety of ENA adapters through feature negotiation with the adapter and upgradable commands set. The ENA driver handles PCI Physical and Virtual ENA functions.

- **Restored vmxnet3 TX data ring.**

  TX data ring has been shown to improve small packet forwarding performance on the vSphere environment.

- **Added vmxnet3 TX L4 checksum offload.**

  Added support for TCP/UDP checksum offload to vmxnet3.

- **Added vmxnet3 TSO support.**

  Added support for TSO to vmxnet3.

- **Added vmxnet3 support for jumbo frames.**

  Added support for linking multi-segment buffers together to handle Jumbo packets.

- **Enabled Virtio 1.0 support.**

  Enabled Virtio 1.0 support for Virtio pmd driver.

- **Supported Virtio for ARM.**

  Enabled Virtio support for ARMv7/v8. Tested for ARM64. Virtio for ARM supports VFIO-noiommu mode only. Virtio can work with other non-x86 architectures as well, like PowerPC.

- **Supported Virtio offload in vhost-user.**

  Added the offload and negotiation of checksum and TSO between vhost-user and vanilla Linux Virtio guest.

- **Added vhost-user live migration support.**

- **Added vhost driver.**

  Added a virtual PMD that wraps `librte_vhost`.

- **Added multicast promiscuous mode support on VF for ixgbe.**

  Added multicast promiscuous mode support for the ixgbe VF driver so all VFs can receive the multicast packets.

  Please note if you want to use this promiscuous mode, you need both PF and VF driver to support it. The reason is that this VF feature is configured in the PF. If you use kernel PF driver and the dpdk VF driver, make sure the kernel PF driver supports VF multicast promiscuous mode. If you use dpdk PF and dpdk VF ensure the PF driver is the same version as the VF.

- **Added support for E-tag on X550.**

  E-tag is defined in 802.1BR - Bridge Port Extension.

  This feature is for the VF, but the settings are on the PF. It means the CLIs should be used on the PF, but some of their effects will be shown on the VF. The forwarding of E-tag packets based on GRP and E-CID_base will have an effect on the PF. Theoretically, the E-tag packets can be forwarded to any pool/queue but normally we'd like to forward the packets to the pools/queues belonging to the VFs. And E-tag insertion and stripping will have an effect on VFs. When a VF receives E-tag packets it should strip the E-tag. When the VF transmits packets, it should insert the E-tag. Both actions can be offloaded.

  When we want to use this E-tag support feature, the forwarding should be enabled to forward the packets received by the PF to the indicated VFs. And insertion and stripping should be enabled for VFs to offload the effort to hardware.

  Features added:

  - Support E-tag offloading of insertion and stripping.
  - Support Forwarding E-tag packets to pools based on GRP and E-CID_base.

- **Added support for VxLAN and NVGRE checksum off-load on X550.**

  - Added support for VxLAN and NVGRE RX/TX checksum off-load on X550. RX/TX checksum off-load is provided on both inner and outer IP header and TCP header.
  - Added functions to support VxLAN port configuration. The default VxLAN port number is 4789 but this can be updated programmatically.

- **Added support for new X550EM_a devices.**

  Added support for new X550EM_a devices and their MAC types, X550EM_a and X550EM_a_vf. Updated the relevant PMD to use the new devices and MAC types.

- **Added x550em_x V2 device support.**

  Added support for x550em_x V2 device. Only x550em_x V1 was supported before. A mask for V1 and V2 is defined and used to support both.

- **Supported link speed auto-negotiation on X550EM_X**

  Normally the auto-negotiation is supported by firmware and software doesn't care about it. But on x550em_x, firmware doesn't support auto-negotiation. As the ports of x550em_x are 10GbE, if we connect the port with a peer which is 1GbE, the link will always be down. We added the support for auto-negotiation by software to avoid this link down issue.

- **Added software-firmware sync on X550EM_a.**

  Added support for software-firmware sync for resource sharing. Use the PHY token, shared between software-firmware for PHY access on X550EM_a.

- **Updated the i40e base driver.**

  The i40e base driver was updated with changes including the following:

  - Use RX control AQ commands to read/write RX control registers.
  - Add new X722 device IDs, and removed X710 one was never used.
  - Expose registers for HASH/FD input set configuring.

- **Enabled PCI extended tag for i40e.**

  Enabled extended tag for i40e by checking and writing corresponding PCI config space bytes, to boost the performance. The legacy method of reading/writing sysfile supported by kernel module igb_uio is now deprecated.

- **Added i40e support for setting mac addresses.**
- **Added dump of i40e registers and EEPROM.**
- **Supported ether type setting of single and double VLAN for i40e**
- **Added VMDQ DCB mode in i40e.**

  Added support for DCB in VMDQ mode to i40e driver.

- **Added i40e VEB switching support.**
- **Added Flow director enhancements in i40e.**
- **Added PF reset event reporting in i40e VF driver.**
- **Added fm10k RX interrupt support.**
- **Optimized fm10k TX.**

  Optimized fm10k TX by freeing multiple mbufs at a time.

- **Handled error flags in fm10k vector RX.**

  Parse error flags in RX descriptor and set error bits in mbuf with vector instructions.

- **Added fm10k FTAG based forwarding support.**
- **Added mlx5 flow director support.**

  Added flow director support (`RTE_FDIR_MODE_PERFECT` and `RTE_FDIR_MODE_PERFECT_MAC_VLAN`).

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 RX VLAN stripping support.**

  Added support for RX VLAN stripping.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 link up/down callbacks.**

  Implemented callbacks to bring link up and down.
- **Added mlx5 support for operation in secondary processes.**

  Implemented TX support in secondary processes (like mlx4).
- **Added mlx5 RX CRC stripping configuration.**

  Until now, CRC was always stripped. It can now be configured.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 optional packet padding by HW.**

  Added an option to make PCI bus transactions rounded to a multiple of a cache line size for better alignment.

  Only available with Mellanox OFED >= 3.2.
- **Added mlx5 TX VLAN insertion support.**

  Added support for TX VLAN insertion.

  Only available with Mellanox OFED >= 3.2.
- **Changed szedata2 driver type from vdev to pdev.**

  Previously szedata2 device had to be added by `--vdev` option. Now szedata2 PMD recognizes the device automatically during EAL initialization.
- **Added szedata2 functions for setting link up/down.**
- **Added szedata2 promiscuous and allmulticast modes.**
- **Added af_packet dynamic removal function.**

  An af_packet device can now be detached using the API, like other PMD devices.
- **Increased number of next hops for LPM IPv4 to 2^24.**

  The `next_hop` field has been extended from 8 bits to 24 bits for IPv4.
- **Added support of SNOW 3G (UEA2 and UIA2) for Intel Quick Assist devices.**

  Enabled support for the SNOW 3G wireless algorithm for Intel Quick Assist devices. Support for cipher-only and hash-only is also provided along with algorithm-chaining operations.

- **Added SNOW3G SW PMD.**

  A new Crypto PMD has been added, which provides SNOW 3G UEA2 ciphering and SNOW3G UIA2 hashing.

- **Added AES GCM PMD.**

  Added new Crypto PMD to support AES-GCM authenticated encryption and authenticated decryption in software.

- **Added NULL Crypto PMD**

  Added new Crypto PMD to support null crypto operations in software.

- **Improved IP Pipeline Application.**

  The following features have been added to ip_pipeline application;

  - Added CPU utilization measurement and idle cycle rate computation.
  - Added link identification support through existing port-mask option or by specifying PCI device in every LINK section in the configuration file.
  - Added load balancing support in passthrough pipeline.

- **Added IPsec security gateway example.**

  Added a new application implementing an IPsec Security Gateway.

11.2 Resolved Issues

11.2.1 Drivers

- **ethdev: Fixed overflow for 100Gbps.**

  100Gbps in Mbps (100000) was exceeding the 16-bit max value of `link_speed` in `rte_eth_link`.

- **ethdev: Fixed byte order consistency between fdir flow and mask.**

  Fixed issue in ethdev library where the structure for setting fdir's mask and flow entry was not consistent in byte ordering.

- **cxgbe: Fixed crash due to incorrect size allocated for RSS table.**

  Fixed a segfault that occurs when accessing part of port 0's RSS table that gets overwritten by subsequent port 1's part of the RSS table due to incorrect size allocated for each entry in the table.

- **cxgbe: Fixed setting wrong device MTU.**

  Fixed an incorrect device MTU being set due to the Ethernet header and CRC lengths being added twice.

- **ixgbe: Fixed zeroed VF mac address.**

  Resolved an issue where the VF MAC address is zeroed out in cases where the VF driver is loaded while the PF interface is down. The solution is to only set it when we get an ACK from the PF.

- **ixgbe: Fixed setting flow director flag twice.**

  Resolved an issue where packets were being dropped when switching to perfect filters mode.

- **ixgbe: Set MDIO speed after MAC reset.**

  The MDIO clock speed must be reconfigured after the MAC reset. The MDIO clock speed becomes invalid, therefore the driver reads invalid PHY register values. The driver now set the MDIO clock speed prior to initializing PHY ops and again after the MAC reset.

- **ixgbe: Fixed maximum number of available TX queues.**

  In IXGBE, the maximum number of TX queues varies depending on the NIC operating mode. This was not being updated in the device information, providing an incorrect number in some cases.

- **i40e: Generated MAC address for each VFs.**

  It generates a MAC address for each VFs during PF host initialization, and keeps the VF MAC address the same among different VF launch.

- **i40e: Fixed failure of reading/writing RX control registers.**

  Fixed i40e issue of failing to read/write rx control registers when under stress with traffic, which might result in application launch failure.

- **i40e: Enabled vector driver by default.**

  Previously, vector driver was disabled by default as it couldn't fill packet type info for l3fwd to work well. Now there is an option for l3fwd to analyze the packet type so the vector driver is enabled by default.

- **i40e: Fixed link info of VF.**

  Previously, the VF's link speed stayed at 10GbE and status always was up. It did not change even when the physical link's status changed. Now this issue is fixed to make VF's link info consistent with physical link.

- **mlx5: Fixed possible crash during initialization.**

  A crash could occur when failing to allocate private device context.

- **mlx5: Added port type check.**

  Added port type check to prevent port initialization on non-Ethernet link layers and to report an error.

- **mlx5: Applied VLAN filtering to broadcast and IPv6 multicast flows.**

  Prevented reception of multicast frames outside of configured VLANs.

- **mlx5: Fixed RX checksum offload in non L3/L4 packets.**

  Fixed report of bad checksum for packets of unknown type.

- **aesni_mb: Fixed wrong return value when creating a device.**

  The `cryptodev_aesni_mb_init()` function was returning the device id of the device created, instead of 0 (on success) that `rte_eal_vdev_init()` expects. This made it impossible to create more than one aesni_mb device from the command line.

- **qat: Fixed AES GCM decryption.**

  Allowed AES GCM on the cryptodev API, but in some cases gave invalid results due to incorrect IV setting.

11.2.2 Libraries

- **hash: Fixed CRC32c hash computation for non multiple of 4 bytes sizes.**

  Fix crc32c hash functions to return a valid crc32c value for data lengths not a multiple of 4 bytes.

- **hash: Fixed hash library to support multi-process mode.**

  Fix hash library to support multi-process mode, using a jump table, instead of storing a function pointer to the key compare function. Multi-process mode only works with the built-in compare functions, however a custom compare function (not in the jump table) can only be used in single-process mode.

- **hash: Fixed return value when allocating an existing hash table.**

  Changed the `rte_hash*_create()` functions to return `NULL` and set `rte_errno` to `EEXIST` when the object name already exists. This is the behavior described in the API documentation in the header file. The previous behavior was to return a pointer to the existing object in that case, preventing the caller from knowing if the object had to be freed or not.

- **lpm: Fixed return value when allocating an existing object.**

  Changed the `rte_lpm*_create()` functions to return `NULL` and set `rte_errno` to `EEXIST` when the object name already exists. This is the behavior described in the API documentation in the header file. The previous behavior was to return a pointer to the existing object in that case, preventing the caller from knowing if the object had to be freed or not.

- **librte_port: Fixed segmentation fault for ring and ethdev writer nodrop.**

  Fixed core dump issue on txq and swq when dropless is set to yes.

11.2.3 Examples

- **l3fwd-power: Fixed memory leak for non-IP packet.**

  Fixed issue in l3fwd-power where, on receiving packets of types other than IPv4 or IPv6, the mbuf was not released, and caused a memory leak.

- **l3fwd: Fixed using packet type blindly.**

  l3fwd makes use of packet type information without querying if devices or PMDs really set it. For those devices that don't set ptypes, add an option to parse it.

- **examples/vhost: Fixed frequent mbuf allocation failure.**

  The vhost-switch often fails to allocate mbuf when dequeue from vring because it wrongly calculates the number of mbufs needed.

11.3 API Changes

- The ethdev statistics counter `imissed` is considered to be independent of `ierrors`. All drivers are now counting the missed packets only once, i.e. drivers will not increment ierrors anymore for missed packets.
- The ethdev structure `rte_eth_dev_info` was changed to support device speed capabilities.
- The ethdev structures `rte_eth_link` and `rte_eth_conf` were changed to support the new link API.
- The functions `rte_eth_dev_udp_tunnel_add` and `rte_eth_dev_udp_tunnel_delete` have been renamed into `rte_eth_dev_udp_tunnel_port_add` and `rte_eth_dev_udp_tunnel_port_delete`.
- The `outer_mac` and `inner_mac` fields in structure `rte_eth_tunnel_filter_conf` are changed from pointer to struct in order to keep code's readability.
- The fields in ethdev structure `rte_eth_fdir_masks` were changed to be in big endian.
- A parameter `vlan_type` has been added to the function `rte_eth_dev_set_vlan_ether_type`.
- The af_packet device init function is no longer public. The device should be attached via the API.
- The LPM `next_hop` field is extended from 8 bits to 24 bits for IPv4 while keeping ABI compatibility.
- A new `rte_lpm_config` structure is used so the LPM library will allocate exactly the amount of memory which is necessary to hold application's rules. The previous ABI is kept for compatibility.
- The prototype for the pipeline input port, output port and table action handlers are updated: the pipeline parameter is added, the packets mask parameter has been either removed or made input-only.

11.4 ABI Changes

- The RETA entry size in `rte_eth_rss_reta_entry64` has been increased from 8-bit to 16-bit.
- The ethdev flow director structure `rte_eth_fdir_flow` structure was changed. New fields were added to extend flow director's input set.
- The cmdline buffer size has been increase from 256 to 512.

11.5 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.3
  librte_acl.so.2
  librte_cfgfile.so.2
+ librte_cmdline.so.2
  librte_distributor.so.1
  librte_eal.so.2
  librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
  librte_kni.so.2
  librte_kvargs.so.1
  librte_lpm.so.2
  librte_mbuf.so.2
  librte_mempool.so.1
  librte_meter.so.1
+ librte_pipeline.so.3
  librte_pmd_bond.so.1
  librte_pmd_ring.so.2
  librte_port.so.2
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
  librte_table.so.2
  librte_timer.so.1
  librte_vhost.so.2
```

11.6 Tested Platforms

1. SuperMicro 1U
   - BIOS: 1.0c
   - Processor: Intel(R) Atom(TM) CPU C2758 @ 2.40GHz
2. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU D-1540 @ 2.00GHz
   - Onboard NIC: Intel(R) X552/X557-AT (2x10G)
     - Firmware-version: 0x800001cf
     - Device ID (PF/VF): 8086:15ad /8086:15a8
   - kernel driver version: 4.2.5 (ixgbe)
3. SuperMicro 1U
   - BIOS: 1.0a
   - Processor: Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz
4. Intel(R) Server board S2600GZ
   - BIOS: SE5C600.86B.02.02.0002.122320131210
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
5. Intel(R) Server board W2600CR

   - BIOS: SE5C600.86B.02.01.0002.082220131453
   - Processor: Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
6. Intel(R) Server board S2600CWT
   - BIOS: SE5C610.86B.01.01.0009.060120151350
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
7. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.01.01.0005.101720141054
   - Processor: Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz
8. Intel(R) Server board S2600WTT
   - BIOS: SE5C610.86B.11.01.0044.090120151156
   - Processor: Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz

11.7 Tested NICs

1. Intel(R) Ethernet Controller X540-AT2
   - Firmware version: 0x80000389
   - Device id (pf): 8086:1528
   - Driver version: 3.23.2 (ixgbe)
2. Intel(R) 82599ES 10 Gigabit Ethernet Controller
   - Firmware version: 0x61bf0001
   - Device id (pf/vf): 8086:10fb / 8086:10ed
   - Driver version: 4.0.1-k (ixgbe)
3. Intel(R) Corporation Ethernet Connection X552/X557-AT 10GBASE-T
   - Firmware version: 0x800001cf
   - Device id (pf/vf): 8086:15ad / 8086:15a8
   - Driver version: 4.2.5 (ixgbe)
4. Intel(R) Ethernet Converged Network Adapter X710-DA4 (4x10G)
   - Firmware version: 5.02 0x80002284
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.26 (i40e)
5. Intel(R) Ethernet Converged Network Adapter X710-DA2 (2x10G)
   - Firmware version: 5.02 0x80002282
   - Device id (pf/vf): 8086:1572 / 8086:154c
   - Driver version: 1.4.25 (i40e)
6. Intel(R) Ethernet Converged Network Adapter XL710-QDA1 (1x40G)

    - Firmware version: 5.02 0x80002281
    - Device id (pf/vf): 8086:1584 / 8086:154c
    - Driver version: 1.4.25 (i40e)
7. Intel(R) Ethernet Converged Network Adapter XL710-QDA2 (2X40G)
    - Firmware version: 5.02 0x80002285
    - Device id (pf/vf): 8086:1583 / 8086:154c
    - Driver version: 1.4.25 (i40e)
8. Intel(R) 82576EB Gigabit Ethernet Controller
    - Firmware version: 1.2.1
    - Device id (pf): 8086:1526
    - Driver version: 5.2.13-k (igb)
9. Intel(R) Ethernet Controller I210
    - Firmware version: 3.16, 0x80000500, 1.304.0
    - Device id (pf): 8086:1533
    - Driver version: 5.2.13-k (igb)
10. Intel(R) Corporation I350 Gigabit Network Connection
    - Firmware version: 1.48, 0x800006e7
    - Device id (pf/vf): 8086:1521 / 8086:1520
    - Driver version: 5.2.13-k (igb)
11. Intel(R) Ethernet Multi-host Controller FM10000
    - Firmware version: N/A
    - Device id (pf/vf): 8086:15d0
    - Driver version: 0.17.0.9 (fm10k)

CHAPTER

TWELVE

DPDK RELEASE 2.2

12.1 New Features

- **Introduce ARMv7 and ARMv8 architectures.**
  - It is now possible to build DPDK for the ARMv7 and ARMv8 platforms.
  - ARMv7 can be tested with virtual PMD drivers.
  - ARMv8 can be tested with virtual and physical PMD drivers.
- **Enabled freeing of ring.**

  A new function `rte_ring_free()` has been added to allow the user to free a ring if it was created with `rte_ring_create()`.
- **Added keepalive support to EAL and example application.**
- **Added experimental cryptodev API**

  The cryptographic processing of packets is provided as a preview with two drivers for:
  - Intel QuickAssist devices
  - Intel AES-NI multi-buffer library

  Due to its experimental state, the API may change without prior notice.
- **Added ethdev APIs for additional IEEE1588 support.**

  Added functions to read, write and adjust system time in the NIC. Added client slave sample application to demonstrate the IEEE1588 functionality.
- **Extended Statistics.**

  Defined an extended statistics naming scheme to store metadata in the name string of each statistic. Refer to the Extended Statistics section of the Programmers Guide for more details.

  Implemented the extended statistics API for the following PMDs:
  - `igb`
  - `igbvf`
  - `i40e`
  - `i40evf`
  - `fm10k`

  - `virtio`

- **Added API in ethdev to retrieve RX/TX queue information.**
  - Added the ability for the upper layer to query RX/TX queue information.
  - Added new fields in `rte_eth_dev_info` to represent information about RX/TX descriptors min/max/align numbers, per queue, for the device.
- **Added RSS dynamic configuration to bonding.**
- **Updated the e1000 base driver.**

  The e1000 base driver was updated with several features including the following:
  - Added new i218 devices
  - Allowed both ULP and EEE in Sx state
  - Initialized 88E1543 (Marvell 1543) PHY
  - Added flags to set EEE advertisement modes
  - Supported inverted format ETrackId
  - Added bit to disable packetbuffer read
  - Added defaults for i210 RX/TX PBSIZE
  - Check more errors for ESB2 init and reset
  - Check more NVM read errors
  - Return code after setting receive address register
  - Removed all NAHUM6LP_HW tags
- **Added e1000 RX interrupt support.**
- **Added igb TSO support for both PF and VF.**
- **Added RSS enhancements to Intel x550 NIC.**
  - Added support for 512 entry RSS redirection table.
  - Added support for per VF RSS redirection table.
- **Added Flow director enhancements on Intel x550 NIC.**
  - Added 2 new flow director modes on x550. One is MAC VLAN mode, the other is tunnel mode.
- **Updated the i40e base driver.**

  The i40e base driver was updated with several changes including the following:
  - Added promiscuous on VLAN support
  - Added a workaround to drop all flow control frames
  - Added VF capabilities to virtual channel interface
  - Added TX Scheduling related AQ commands
  - Added additional PCTYPES supported for FortPark RSS
  - Added parsing for CEE DCBX TLVs

  - Added FortPark specific registers
  - Added AQ functions to handle RSS Key and LUT programming
  - Increased PF reset max loop limit

- **Added i40e vector RX/TX.**
- **Added i40e RX interrupt support.**
- **Added i40e flow control support.**
- **Added DCB support to i40e PF driver.**
- **Added RSS/FD input set granularity on Intel X710/XL710.**
- **Added different GRE key length for input set on Intel X710/XL710.**
- **Added flow director support in i40e VF.**
- **Added i40e support of early X722 series.**

  Added early X722 support, for evaluation only, as the hardware is alpha.

- **Added fm10k vector RX/TX.**
- **Added fm10k TSO support for both PF and VF.**
- **Added fm10k VMDQ support.**
- **New NIC Boulder Rapid support.**

  Added support for the Boulder Rapid variant of Intel's fm10k NIC family.

- **Enhanced support for the Chelsio CXGBE driver.**
  - Added support for Jumbo Frames.
  - Optimized forwarding performance for Chelsio T5 40GbE cards.
- **Improved enic TX packet rate.**

  Reduced frequency of TX tail pointer updates to the NIC.

- **Added support for link status interrupts in mlx4.**
- **Added partial support (TX only) for secondary processes in mlx4.**
- **Added support for Mellanox ConnectX-4 adapters (mlx5).**

  The mlx5 poll-mode driver implements support for Mellanox ConnectX-4 EN and Mellanox ConnectX-4 Lx EN families of 10/25/40/50/100 Gb/s adapters.

  Like mlx4, this PMD is only available for Linux and is disabled by default due to external dependencies (libibverbs and libmlx5).

- **Added driver for Netronome nfp-6xxx card.**

  Support for using Netronome nfp-6xxx with PCI VFs.

- **Added virtual szedata2 driver for COMBO cards.**

  Added virtual PMD for COMBO-100G and COMBO-80G cards. PMD is disabled in default configuration.

- **Enhanced support for virtio driver.**

  - Virtio ring layout optimization (fixed avail ring)
  - Vector RX
  - Simple TX

- **Added vhost-user multiple queue support.**
- **Added port hotplug support to vmxnet3.**
- **Added port hotplug support to xenvirt.**
- **Added ethtool shim and sample application.**
- **Added experimental performance thread example application.**

  The new sample application demonstrates L3 forwarding with different threading models: pthreads, cgroups, or lightweight threads. The example includes a simple cooperative scheduler.

  Due to its experimental state this application may change without notice. The application is supported only for Linux x86_64.

- **Enhancements to the IP pipeline application.**

  The following features have been added to the `ip_pipeline` application;

  - Added Multiple Producers/Multiple Consumers (MPSC) and fragmentation/reassembly support to software rings.
  - Added a dynamic pipeline reconfiguration feature that allows binding a pipeline to other threads at runtime using CLI commands.
  - Added enable/disable of `promisc` mode from `ip_pipeline` configuration file.
  - Added check on RX queues and TX queues of each link whether they are used correctly in the `ip_pipeline` configuration file.
  - Added flow id parameters to the flow-classification table entries.
  - Added more functions to the routing pipeline: ARP table enable/disable, Q-in-Q and MPLS encapsulation, add color (traffic-class for QoS) to the MPLS tag.
  - Added flow-actions pipeline for traffic metering/marking (for e.g. Two Rate Three Color Marker (trTCM)), policer etc.
  - Modified the pass-through pipeline's actions-handler to implement a generic approach to extract fields from the packet's header and copy them to packet metadata.

12.2 Resolved Issues

12.2.1 EAL

- **eal/linux: Fixed epoll timeout.**

  Fixed issue where the `rte_epoll_wait()` function didn't return when the underlying call to `epoll_wait()` timed out.

12.2.2 Drivers

- **e1000/base: Synchronize PHY interface on non-ME systems.**

  On power up, the MAC - PHY interface needs to be set to PCIe, even if the cable is disconnected. In ME systems, the ME handles this on exit from the Sx (Sticky mode) state. In non-ME, the driver handles it. Added a check for non-ME system to the driver code that handles it.

- **e1000/base: Increased timeout of reset check.**

  Previously, in `check_reset_block` RSPCIPHY was polled for 100 ms before determining that the ME veto was set. This was not enough and it was increased to 300 ms.

- **e1000/base: Disabled IPv6 extension header parsing on 82575.**

  Disabled IPv6 options as per hardware limitation.

- **e1000/base: Prevent ULP flow if cable connected.**

  Enabling ULP on link down when the cable is connected caused an infinite loop of link up/down indications in the NDIS driver. The driver now enables ULP only when the cable is disconnected.

- **e1000/base: Support different EEARBC for i210.**

  EEARBC has changed on i210. It means EEARBC has a different address on i210 than on other NICs. So, add a new entity named EEARBC_I210 to the register list and make sure the right one is being used on i210.

- **e1000/base: Fix K1 configuration.**

  Added fix for the following updates to the K1 configurations: TX idle period for entering K1 should be 128 ns. Minimum TX idle period in K1 should be 256 ns.

- **e1000/base: Fix link detect flow.**

  Fix link detect flow in case where auto-negotiate is not enabled, by calling `e1000_setup_copper_link_generic` instead of `e1000_phy_setup_autoneg`.

- **e1000/base: Fix link check for i354 M88E1112 PHY.**

  The `e1000_check_for_link_media_swap()` function is supposed to check PHY page 0 for copper and PHY page 1 for “other” (fiber) links. The driver switched back from page 1 to page 0 too soon, before `e1000_check_for_link_82575()` is executed and was never finding the link on the fiber (other).

  If the link is copper, as the M88E1112 page address is set to 1, it should be set back to 0 before checking this link.

- **e1000/base: Fix beacon duration for i217.**

  Fix for I217 Packet Loss issue - The Management Engine sets the FEXTNVM4 Beacon Duration incorrectly. This fix ensures that the correct value will always be set. Correct value for this field is 8 usec.

- **e1000/base: Fix TIPG for non 10 half duplex mode.**

  TIPG value is increased when setting speed to 10 half duplex to prevent packet loss. However, it was never decreased again when speed changed. This caused performance issues in the NDIS driver. Fix this to restore TIPG to default value on non 10 half duplex.

- **e1000/base: Fix reset of DH89XXCC SGMII.**

  For DH89XXCC_SGMII, a write flush leaves registers of this device trashed (0xFFFFFFFF). Add check for this device.

  Also, after both Port SW Reset and Device Reset case, the platform should wait at least 3ms before reading any registers. Remove this condition since waiting is conditionally executed only for Device Reset.

- **e1000/base: Fix redundant PHY power down for i210.**

  Bit 11 of PHYREG 0 is used to power down PHY. The use of PHYREG 16 is no longer necessary.

- **e1000/base: fix jumbo frame CRC failures.**

  Change the value of register 776.20[11:2] for jumbo mode from 0x1A to 0x1F. This is to enlarge the gap between read and write pointers in the TX FIFO.

- **e1000/base: Fix link flap on 82579.**

  Several customers have reported a link flap issue on 82579. The symptoms are random and intermittent link losses when 82579 is connected to specific switches. the Issue was root caused as an inter-operability problem between the NIC and at least some Broadcom PHYs in the Energy Efficient Ethernet wake mechanism.

  To fix the issue, we are disabling the Phase Locked Loop shutdown in 100M Low Power Idle. This solution will cause an increase of power in 100M EEE link. It may cost an additional 28mW in this specific mode.

- **igb: Fixed IEEE1588 frame identification in I210.**

  Fixed issue where the flag `PKT_RX_IEEE1588_PTP` was not being set in the Intel I210 NIC, as the EtherType in RX descriptor is in bits 8:10 of Packet Type and not in the default bits 0:2.

- **igb: Fixed VF start with PF stopped.**

  VF needs the PF interrupt support initialized even if not started.

- **igb: Fixed VF MAC address when using with DPDK PF.**

  Assign a random MAC address in VF when not assigned by PF.

- **igb: Removed CRC bytes from byte counter statistics.**

- **ixgbe: Fixed issue with X550 DCB.**

  Fixed a DCB issue with x550 where for 8 TCs (Traffic Classes), if a packet with user priority 6 or 7 was injected to the NIC, then the NIC would only put 3 packets into the queue. There was also a similar issue for 4 TCs.

- **ixgbe: Removed burst size restriction of vector RX.**

  Fixed issue where a burst size less than 32 didn't receive anything.

- **ixgbe: Fixed VF start with PF stopped.**

  VF needs the PF interrupt support initialized even if not started.

- **ixgbe: Fixed TX hang when RS distance exceeds HW limit.**

Fixed an issue where the TX queue can hang when a lot of highly fragmented packets have to be sent. As part of that fix, `tx_rs_thresh` for ixgbe PMD is not allowed to be greater then to 32 to comply with HW restrictions.

- **ixgbe: Fixed rx error statistic counter.**

  Fixed an issue that the rx error counter of ixgbe was not accurate. The mac short packet discard count (mspdc) was added to the counter. Mac local faults and mac remote faults are removed as they do not count packets but errors, and jabber errors were removed as they are already accounted for by the CRC error counter. Finally the XEC (l3 / l4 checksum error) counter was removed due to errata, see commit 256ff05a9cae for details.

- **ixgbe: Removed CRC bytes from byte counter statistics.**

- **i40e: Fixed base driver allocation when not using first numa node.**

  Fixed i40e issue that occurred when a DPDK application didn't initialize ports if memory wasn't available on socket 0.

- **i40e: Fixed maximum of 64 queues per port.**

  Fixed an issue in i40e where it would not support more than 64 queues per port, even though the hardware actually supports it. The real number of queues may vary, as long as the total number of queues used in PF, VFs, VMDq and FD does not exceeds the hardware maximum.

- **i40e: Fixed statistics of packets.**

  Added discarding packets on VSI to the stats and rectify the old statistics.

- **i40e: Fixed issue of not freeing memzone.**

  Fixed an issue of not freeing a memzone in the call to free the memory for adminq DMA.

- **i40e: Removed CRC bytes from byte counter statistics.**

- **mlx: Fixed driver loading.**

  The mlx drivers were unable to load when built as a shared library, due to a missing symbol in the mempool library.

- **mlx4: Performance improvements.**

  Fixed bugs in TX and RX flows that improves mlx4 performance.

- **mlx4: Fixed TX loss after initialization.**

- **mlx4: Fixed scattered TX with too many segments.**

- **mlx4: Fixed memory registration for indirect mbuf data.**

- **vhost: Fixed Qemu shutdown.**

  Fixed issue with libvirt `virsh destroy` not killing the VM.

- **virtio: Fixed crash after changing link state.**

  Fixed IO permission in the interrupt handler.

- **virtio: Fixed crash when releasing queue.**

  Fixed issue when releasing null control queue.

12.2.3 Libraries

- **hash: Fixed memory allocation of Cuckoo Hash key table.**

  Fixed issue where an incorrect Cuckoo Hash key table size could be calculated limiting the size to 4GB.

- **hash: Fixed incorrect lookup if key is all zero.**

  Fixed issue in hash library that occurred if an all zero key was not added to the table and the key was looked up, resulting in an incorrect hit.

- **hash: Fixed thread scaling by reducing contention.**

  Fixed issue in the hash library where, using multiple cores with hardware transactional memory support, thread scaling did not work, due to the global ring that is shared by all cores.

12.2.4 Examples

- **l3fwd: Fixed crash with IPv6.**
- **vhost_xen: Fixed compile error.**

12.2.5 Other

- This release drops compatibility with Linux kernel 2.6.33. The minimum kernel requirement is now 2.6.34.

12.3 Known Issues

- Some drivers do not fill in the packet type when receiving. As the l3fwd example application requires this info, the i40e vector driver must be disabled to benefit of the packet type with i40e.
- Some (possibly all) VF drivers (e.g. i40evf) do not handle any PF reset events/requests in the VF driver. This means that the VF driver may not work after a PF reset in the host side. The workaround is to avoid triggering any PF reset events/requests on the host side.
- 100G link report support is missing.
- **Mellanox PMDs (mlx4 & mlx5):**
  - PMDs do not support CONFIG_RTE_BUILD_COMBINE_LIBS and CONFIG_RTE_BUILD_SHARED_LIB simultaneously.
  - There is performance degradation for small packets when the PMD is compiled with `SGE_WR_N = 4` compared to the performance when `SGE_WR_N = 1`. If scattered packets are not used it is recommended to compile the PMD with `SGE_WR_N = 1`.
  - When a Multicast or Broadcast packet is sent to the SR-IOV mlx4 VF, it is returned back to the port.
  - PMDs report “bad” L4 checksum when IP packet is received.

  - mlx5 PMD reports “bad” checksum although the packet has “good” checksum. Will be fixed in upcoming MLNX_OFED release.

12.4 API Changes

- The deprecated flow director API is removed. It was replaced by `rte_eth_dev_filter_ctrl()`.
- The `dcb_queue` is renamed to `dcb_tc` in following dcb configuration structures: `rte_eth_dcb_rx_conf`, `rte_eth_dcb_tx_conf`, `rte_eth_vmdq_dcb_conf`, `rte_eth_vmdq_dcb_tx_conf`.
- The `rte_eth_rx_queue_count()` function now returns “int” instead of “uint32_t” to allow the use of negative values as error codes on return.
- The function `rte_eal_pci_close_one()` is removed. It was replaced by `rte_eal_pci_detach()`.
- The deprecated ACL API `ipv4vlan` is removed.
- The deprecated hash function `rte_jhash2()` is removed. It was replaced by `rte_jhash_32b()`.
- The deprecated KNI functions are removed: `rte_kni_create()`, `rte_kni_get_port_id()` and `rte_kni_info_get()`.
- The deprecated ring PMD functions are removed: `rte_eth_ring_pair_create()` and `rte_eth_ring_pair_attach()`.
- The devargs union field `virtual` is renamed to `virt` for C++ compatibility.

12.5 ABI Changes

- The EAL and ethdev structures `rte_intr_handle` and `rte_eth_conf` were changed to support RX interrupt. This was already included in 2.1 under the `CONFIG_RTE_NEXT_ABI` #define.
- The ethdev flow director entries for SCTP were changed. This was already included in 2.1 under the `CONFIG_RTE_NEXT_ABI` #define.
- The ethdev flow director structure `rte_eth_fdir_flow_ext` structure was changed. New fields were added to support flow director filtering in VF.
- The size of the ethdev structure `rte_eth_hash_filter_info` is changed by adding a new element `rte_eth_input_set_conf` in a union.
- New fields `rx_desc_lim` and `tx_desc_lim` are added into `rte_eth_dev_info` structure.
- For debug builds, the functions `rte_eth_rx_burst()`, `rte_eth_tx_burst()` `rte_eth_rx_descriptor_done()` and `rte_eth_rx_queue_count()` will no longer be separate functions in the DPDK libraries. Instead, they will only be present in the `rte_ethdev.h` header file.
- The maximum number of queues per port `CONFIG_RTE_MAX_QUEUES_PER_PORT` is increased to 1024.

- The mbuf structure was changed to support the unified packet type. This was already included in 2.1 under the `CONFIG_RTE_NEXT_ABI` #define.
- The dummy malloc library is removed. The content was moved into EAL in 2.1.
- The LPM structure is changed. The deprecated field `mem_location` is removed.
- librte_table LPM: A new parameter to hold the table name will be added to the LPM table parameter structure.
- librte_table hash: The key mask parameter is added to the hash table parameter structure for 8-byte key and 16-byte key extendable bucket and LRU tables.
- librte_port: Macros to access the packet meta-data stored within the packet buffer has been adjusted to cover the packet mbuf structure.
- librte_cfgfile: Allow longer names and values by increasing the constants `CFG_NAME_LEN` and `CFG_VALUE_LEN` to 64 and 256 respectively.
- vhost: a new field enabled is added to the `vhost_virtqueue` structure.
- vhost: a new field `virt_qp_nb` is added to `virtio_net` structure, and the `virtqueue` field is moved to the end of virtio_net structure.
- vhost: a new operation `vring_state_changed` is added to `virtio_net_device_ops` structure.
- vhost: a few spaces are reserved both at `vhost_virtqueue` and `virtio_net` structure for future extension.

12.6 Shared Library Versions

The libraries prepended with a plus sign were incremented in this version.

```
+ libethdev.so.2
+ librte_acl.so.2
+ librte_cfgfile.so.2
  librte_cmdline.so.1
  librte_distributor.so.1
+ librte_eal.so.2
+ librte_hash.so.2
  librte_ip_frag.so.1
  librte_ivshmem.so.1
  librte_jobstats.so.1
+ librte_kni.so.2
  librte_kvargs.so.1
+ librte_lpm.so.2
+ librte_mbuf.so.2
  librte_mempool.so.1
  librte_meter.so.1
+ librte_pipeline.so.2
  librte_pmd_bond.so.1
+ librte_pmd_ring.so.2
+ librte_port.so.2
  librte_power.so.1
  librte_reorder.so.1
  librte_ring.so.1
  librte_sched.so.1
+ librte_table.so.2
  librte_timer.so.1
+ librte_vhost.so.2
```

CHAPTER

THIRTEEN

DPDK RELEASE 2.1

13.1 New Features

- **Enabled cloning of indirect mbufs.**

  This feature removes a limitation of `rte_pktmbuf_attach()` which generated the warning: “mbuf we’re attaching to must be direct”.

  Now, when attaching to an indirect mbuf it is possible to:

  - Copy all relevant fields (address, length, offload, ...) as before.
  - Get the pointer to the mbuf that embeds the data buffer (direct mbuf), and increase the reference counter.

  When detaching the mbuf, we can now retrieve this direct mbuf as the pointer is determined from the buffer address.

- **Extended packet type support.**

  In previous releases mbuf packet types were indicated by 6 bits in the `ol_flags`. This was not enough for some supported NICs. For example i40e hardware can recognize more than 150 packet types. Not being able to identify these additional packet types limits access to hardware offload capabilities

  So an extended “unified” packet type was added to support all possible PMDs. The 16 bit packet_type in the mbuf structure was changed to 32 bits and used for this purpose.

  To avoid breaking ABI compatibility, the code changes for this feature are enclosed in a `RTE_NEXT_ABI` ifdef. This is enabled by default but can be turned off for ABI compatibility with DPDK R2.0.

- **Reworked memzone to be allocated by malloc and also support freeing.**

  In the memory hierarchy, memsegs are groups of physically contiguous hugepages, memzones are slices of memsegs, and malloc slices memzones into smaller memory chunks.

  This feature modifies `malloc()` so it partitions memsegs instead of memzones. Now memzones allocate their memory from the malloc heap.

  Backward compatibility with API and ABI are maintained.

  This allow memzones, and any other structure based on memzones, for example mempools, to be freed. Currently only the API from freeing memzones is supported.

- **Interrupt mode PMD.**

  This feature introduces a low-latency one-shot RX interrupt into DPDK. It also adds a polling and interrupt mode switch control example.

  DPDK userspace interrupt notification and handling mechanism is based on UIO/VFIO with the following limitations:

  - Per queue RX interrupt events are only allowed in VFIO which supports multiple MSI-X vectors.
  - In UIO, the RX interrupt shares the same vector with other interrupts. When the RX interrupt and LSC interrupt are both enabled, only the former is available.
  - RX interrupt is only implemented for the linuxapp target.
  - The feature is only currently enabled for tow PMDs: ixgbe and igb.

- **Packet Framework enhancements.**

  Several enhancements were made to the Packet Framework:

  - A new configuration file syntax has been introduced for IP pipeline applications. Parsing of the configuration file is changed.
  - Implementation of the IP pipeline application is modified to make it more structured and user friendly.
  - Implementation of the command line interface (CLI) for each pipeline type has been moved to the separate compilation unit. Syntax of pipeline CLI commands has been changed.
  - Initialization of IP pipeline is modified to match the new parameters structure.
  - New implementation of pass-through pipeline, firewall pipeline, routing pipeline, and flow classification has been added.
  - Master pipeline with CLI interface has been added.
  - Added extended documentation of the IP Pipeline.

- **Added API for IEEE1588 timestamping.**

  This feature adds an ethdev API to enable, disable and read IEEE1588/802.1AS PTP timestamps from devices that support it. The following functions were added:

  - `rte_eth_timesync_enable()`
  - `rte_eth_timesync_disable()`
  - `rte_eth_timesync_read_rx_timestamp()`
  - `rte_eth_timesync_read_tx_timestamp()`

  The “ieee1588” forwarding mode in testpmd was also refactored to demonstrate the new API.

- **Added multicast address filtering.**

  Added multicast address filtering via a new ethdev function `set_mc_addr_list()`.

  This overcomes a limitation in previous releases where the receipt of multicast packets on a given port could only be enabled by invoking the `rte_eth_allmulticast_enable()` function. This method did not work for VFs

in SR-IOV architectures when the host PF driver does not allow these operation on VFs. In such cases, joined multicast addresses had to be added individually to the set of multicast addresses that are filtered by the [VF] port.

- **Added Flow Director extensions.**

  Several Flow Director extensions were added such as:

  - Support for RSS and Flow Director hashes in vector RX.
  - Added Flow Director for L2 payload.

- **Added RSS hash key size query per port.**

  This feature supports querying the RSS hash key size of each port. A new field `hash_key_size` has been added in the `rte_eth_dev_info` struct for storing hash key size in bytes.

- **Added userspace ethtool support.**

  Added userspace ethtool support to provide a familiar interface for applications that manage devices via kernel-space `ethtool_op` and `net_device_op`.

  The initial implementation focuses on operations that can be implemented through existing `netdev` APIs. More operations will be supported in later releases.

- **Updated the ixgbe base driver.**

  The ixgbe base driver was updated with several changes including the following:

  - Added a new 82599 device id.
  - Added new X550 PHY ids.
  - Added SFP+ dual-speed support.
  - Added wait helper for X550 IOSF accesses.
  - Added X550em features.
  - Added X557 PHY LEDs support.
  - Commands for flow director.
  - Issue firmware command when resetting X550em.

  See the git log for full details of the ixgbe/base changes.

- **Added additional hotplug support.**

  Port hotplug support was added to the following PMDs:

  - e1000/igb.
  - ixgbe.
  - i40e.
  - fm10k.
  - ring.
  - bonding.
  - virtio.

Port hotplug support was added to BSD.

- **Added ixgbe LRO support.**

  Added LRO support for x540 and 82599 devices.

- **Added extended statistics for ixgbe.**

  Implemented `xstats_get()` and `xstats_reset()` in dev_ops for ixgbe to expose detailed error statistics to DPDK applications.

  These will be implemented for other PMDs in later releases.

- **Added proc_info application.**

  Created a new `proc_info` application, by refactoring the existing `dump_cfg` application, to demonstrate the usage of retrieving statistics, and the new extended statistics (see above), for DPDK interfaces.

- **Updated the i40e base driver.**

  The i40e base driver was updated with several changes including the following:

  - Support for building both PF and VF driver together.
  - Support for CEE DCBX on recent firmware versions.
  - Replacement of `i40e_debug_read_register()`.
  - Rework of `i40e_hmc_get_object_va`.
  - Update of shadow RAM read/write functions.
  - Enhancement of polling NVM semaphore.
  - Enhancements on adminq init and sending asq command.
  - Update of get/set LED functions.
  - Addition of AOC phy types to case statement in get_media_type.
  - Support for iSCSI capability.
  - Setting of FLAG_RD when sending driver version to FW.

  See the git log for full details of the i40e/base changes.

- **Added support for port mirroring in i40e.**

  Enabled mirror functionality in the i40e driver.

- **Added support for i40e double VLAN, QinQ, stripping and insertion.**

  Added support to the i40e driver for offloading double VLAN (QinQ) tags to the mbuf header, and inserting double vlan tags by hardware to the packets to be transmitted. Added a new field `vlan_tci_outer` in the `rte_mbuf` struct, and new flags in `ol_flags` to support this feature.

- **Added fm10k promiscuous mode support.**

  Added support for promiscuous/allmulticast enable and disable in the fm10k PF function. VF is not supported yet.

- **Added fm10k jumbo frame support.**

  Added support for jumbo frame less than 15K in both VF and PF functions in the fm10k pmd.

- **Added fm10k mac vlan filtering support.**

  Added support for the fm10k MAC filter, only available in PF. Updated the VLAN filter to add/delete one static entry in the MAC table for each combination of VLAN and MAC address.

- **Added support for the Broadcom bnx2x driver.**

  Added support for the Broadcom NetXtreme II bnx2x driver. It is supported only on Linux 64-bit and disabled by default.

- **Added support for the Chelsio CXGBE driver.**

  Added support for the CXGBE Poll Mode Driver for the Chelsio Terminator 5 series of 10G/40G adapters.

- **Enhanced support for Mellanox ConnectX-3 driver (mlx4).**
  - Support Mellanox OFED 3.0.
  - Improved performance for both RX and TX operations.
  - Better link status information.
  - Outer L3/L4 checksum offload support.
  - Inner L3/L4 checksum offload support for VXLAN.

- **Enabled VMXNET3 vlan filtering.**

  Added support for the VLAN filter functionality of the VMXNET3 interface.

- **Added support for vhost live migration.**

  Added support to allow live migration of vhost. Without this feature, qemu will report the following error: “migrate: Migration disabled: vhost lacks VHOST_F_LOG_ALL feature”.

- **Added support for pcap jumbo frames.**

  Extended the PCAP PMD to support jumbo frames for RX and TX.

- **Added support for the TILE-Gx architecture.**

  Added support for the EZchip TILE-Gx family of SoCs.

- **Added hardware memory transactions/lock elision for x86.**

  Added the use of hardware memory transactions (HTM) on fast-path for rwlock and spinlock (a.k.a. lock elision). The methods are implemented for x86 using Restricted Transactional Memory instructions (Intel(r) Transactional Synchronization Extensions). The implementation fall-backs to the normal rwlock if HTM is not available or memory transactions fail. This is not a replacement for all rwlock usages since not all critical sections protected by locks are friendly to HTM. For example, an attempt to perform a HW I/O operation inside a hardware memory transaction always aborts the transaction since the CPU is not able to roll-back should the transaction fail. Therefore, hardware transactional locks are not advised to be used around `rte_eth_rx_burst()` and `rte_eth_tx_burst()` calls.

- **Updated Jenkins Hash function**

  Updated the version of the Jenkins Hash (jhash) function used in DPDK from the 1996 version to the 2006 version. This gives up to 35% better performance, compared to the original one.

  Note, the hashes generated by the updated version differ from the hashes generated by the previous version.

- **Added software implementation of the Toeplitz RSS hash**

  Added a software implementation of the Toeplitz hash function used by RSS. It can be used either for packet distribution on a single queue NIC or for simulating RSS computation on a specific NIC (for example after GRE header de-encapsulation).

- **Replaced the existing hash library with a Cuckoo hash implementation.**

  Replaced the existing hash library with another approach, using the Cuckoo Hash method to resolve collisions (open addressing). This method pushes items from a full bucket when a new entry must be added to it, storing the evicted entry in an alternative location, using a secondary hash function.

  This gives the user the ability to store more entries when a bucket is full, in comparison with the previous implementation.

  The API has not been changed, although new fields have been added in the `rte_hash` structure, which has been changed to internal use only.

  The main change when creating a new table is that the number of entries per bucket is now fixed, so its parameter is ignored now (it is still there to maintain the same parameters structure).

  Also, the maximum burst size in lookup_burst function hash been increased to 64, to improve performance.

- **Optimized KNI RX burst size computation.**

  Optimized KNI RX burst size computation by avoiding checking how many entries are in `kni->rx_q` prior to actually pulling them from the fifo.

- **Added KNI multicast.**

  Enabled adding multicast addresses to KNI interfaces by adding an empty callback for `set_rx_mode` (typically used for setting up hardware) so that the ioctl succeeds. This is the same thing as the Linux tap interface does.

- **Added cmdline polling mode.**

  Added the ability to process console input in the same thread as packet processing by using the `poll()` function.

- **Added VXLAN Tunnel End point sample application.**

  Added a Tunnel End point (TEP) sample application that simulates a VXLAN Tunnel Endpoint (VTEP) termination in DPDK. It is used to demonstrate the offload and filtering capabilities of Intel XL710 10/40 GbE NICsfor VXLAN packets.

- **Enabled combining of the ‘‘-m‘‘ and ‘‘–no-huge‘‘ EAL options.**

  Added option to allow combining of the `-m` and `--no-huge` EAL command line options.

This allows user application to run as non-root but with higher memory allocations, and removes a constraint on `--no-huge` mode being limited to 64M.

13.2 Resolved Issues

- **acl: Fix ambiguity between test rules.**

  Some test rules had equal priority for the same category. That could cause an ambiguity in building the trie and test results.

- **acl: Fix invalid rule wildness calculation for bitmask field type.**
- **acl: Fix matching rule.**
- **acl: Fix unneeded trie splitting for subset of rules.**

  When rebuilding a trie for limited rule-set, don't try to split the rule-set even further.

- **app/testpmd: Fix crash when port id out of bound.**

  Fixed issues in testpmd where using a port greater than 32 would cause a seg fault.

  Fixes: edab33b1c01d ("app/testpmd: support port hotplug")

- **app/testpmd: Fix reply to a multicast ICMP request.**

  Set the IP source and destination addresses in the IP header of the ICMP reply.

- **app/testpmd: fix MAC address in ARP reply.**

  Fixed issue where in the `icmpecho` forwarding mode, ARP replies from testpmd contain invalid zero-filled MAC addresses.

  Fixes: 31db4d38de72 ("net: change arp header struct declaration")

- **app/testpmd: fix default flow control values.**

  Fixes: 422a20a4e62d ("app/testpmd: fix uninitialized flow control variables")

- **bonding: Fix crash when stopping inactive slave.**
- **bonding: Fix device initialization error handling.**
- **bonding: Fix initial link status of slave.**

  On Fortville NIC, link status change interrupt callback was not executed when slave in bonding was (re-)started.

- **bonding: Fix socket id for LACP slave.**

  Fixes: 46fb43683679 ("bond: add mode 4")

- **bonding: Fix device initialization error handling.**
- **cmdline: Fix small memory leak.**

  A function in `cmdline.c` had a return that did not free the buf properly.

- **config: Enable same drivers options for Linux and BSD.**

  Enabled vector ixgbe and i40e bulk alloc for BSD as it is already done for Linux.

Fixes: 304caba12643 (“config: fix bsd options”) Fixes: 0ff3324da2eb (“ixgbe: rework vector pmd following mbuf changes”)

- **devargs: Fix crash on failure.**

  This problem occurred when passing an invalid PCI id to the blacklist API in devargs.

- **e1000/i40e: Fix descriptor done flag with odd address.**

- **e1000/igb: fix ieee1588 timestamping initialization.**

  Fixed issue with e1000 ieee1588 timestamp initialization. On initialization the IEEE1588 functions read the system time to set their timestamp. However, on some 1G NICs, for example, i350, system time is disabled by default and the IEEE1588 timestamp was always 0.

- **eal/bsd: Fix inappropriate header guards.**

- **eal/bsd: Fix virtio on FreeBSD.**

  Closing the `/dev/io` fd caused a SIGBUS in inb/outb instructions as the process lost the IOPL privileges once the fd is closed.

  Fixes: 8a312224bcde (“eal/bsd: fix fd leak”)

- **eal/linux: Fix comments on vfio MSI.**

- **eal/linux: Fix irq handling with igb_uio.**

  Fixed an issue where the introduction of `uio_pci_generic` broke interrupt handling with igb_uio.

  Fixes: c112df6875a5 (“eal/linux: toggle interrupt for uio_pci_generic”)

- **eal/linux: Fix numa node detection.**

- **eal/linux: Fix socket value for undetermined numa node.**

  Sets zero as the default value of pci device numa_node if the socket could not be determined. This provides the same default value as FreeBSD which has no NUMA support, and makes the return value of `rte_eth_dev_socket_id()` be consistent with the API description.

- **eal/ppc: Fix cpu cycle count for little endian.**

  On IBM POWER8 PPC64 little endian architecture, the definition of tsc union will be different. This fix enables the right output from `rte_rdtsc()`.

- **ethdev: Fix check of threshold for TX freeing.**

  Fixed issue where the parameter to `tx_free_thresh` was not consistent between the drivers.

- **ethdev: Fix crash if malloc of user callback fails.**

  If `rte_zmalloc()` failed in `rte_eth_dev_callback_register` then the NULL pointer would be dereferenced.

- **ethdev: Fix illegal port access.**

  To obtain a detachable flag, `pci_drv` is accessed in `rte_eth_dev_is_detachable()`. However `pci_drv` is only valid if port is enabled. Fixed by checking `rte_eth_dev_is_valid_port()` first.

- **ethdev: Make tables const.**
- **ethdev: Rename and extend the mirror type.**
- **examples/distributor: Fix debug macro.**

  The macro to turn on additional debug output when the app was compiled with `-DDEBUG` was broken.

  Fixes: 07db4a975094 ("examples/distributor: new sample app")
- **examples/kni: Fix crash on exit.**
- **examples/vhost: Fix build with debug enabled.**

  Fixes: 72ec8d77ac68 ("examples/vhost: rework duplicated code")
- **fm10k: Fix RETA table initialization.**

  The fm10k driver has 128 RETA entries in 32 registers, but it only initialized the first 32 when doing multiple RX queue configurations. This fix initializes all 128 entries.
- **fm10k: Fix RX buffer size.**
- **fm10k: Fix TX multi-segment frame.**
- **fm10k: Fix TX queue cleaning after start error.**
- **fm10k: Fix Tx queue cleaning after start error.**
- **fm10k: Fix default mac/vlan in switch.**
- **fm10k: Fix interrupt fault handling.**
- **fm10k: Fix jumbo frame issue.**
- **fm10k: Fix mac/vlan filtering.**
- **fm10k: Fix maximum VF number.**
- **fm10k: Fix maximum queue number for VF.**

  Both PF and VF shared code in function `fm10k_stats_get()`. The function worked with PF, but had problems with VF since it has less queues than PF.

  Fixes: a6061d9e7075 ("fm10k: register PF driver")
- **fm10k: Fix queue disabling.**
- **fm10k: Fix switch synchronization.**
- **i40e/base: Fix error handling of NVM state update.**
- **i40e/base: Fix hardware port number for pass-through.**
- **i40e/base: Rework virtual address retrieval for lan queue.**
- **i40e/base: Update LED blinking.**
- **i40e/base: Workaround for PHY type with firmware < 4.4.**
- **i40e: Disable setting of PHY configuration.**
- **i40e: Fix SCTP flow director.**

- **i40e: Fix check of descriptor done flag.**

  Fixes: 4861cde46116 ("i40e: new poll mode driver") Fixes: 05999aab4ca6 ("i40e: add or delete flow director")

- **i40e: Fix condition to get VMDQ info.**
- **i40e: Fix registers access from big endian CPU.**
- **i40evf: Clear command when error occurs.**
- **i40evf: Fix RSS with less RX queues than TX queues.**
- **i40evf: Fix crash when setup TX queues.**
- **i40evf: Fix jumbo frame support.**
- **i40evf: Fix offload capability flags.**

  Added checksum offload capability flags which have already been supported for a long time.

- **ivshmem: Fix crash in corner case.**

  Fixed issues where depending on the configured segments it was possible to hit a segmentation fault as a result of decrementing an unsigned index with value 0.

  Fixes: 40b966a211ab ("ivshmem: library changes for mmaping using ivshmem")

- **ixgbe/base: Fix SFP probing.**
- **ixgbe/base: Fix TX pending clearing.**
- **ixgbe/base: Fix X550 CS4227 address.**
- **ixgbe/base: Fix X550 PCIe master disabling.**
- **ixgbe/base: Fix X550 check.**
- **ixgbe/base: Fix X550 init early return.**
- **ixgbe/base: Fix X550 link speed.**
- **ixgbe/base: Fix X550em CS4227 speed mode.**
- **ixgbe/base: Fix X550em SFP+ link stability.**
- **ixgbe/base: Fix X550em UniPHY link configuration.**
- **ixgbe/base: Fix X550em flow control for KR backplane.**
- **ixgbe/base: Fix X550em flow control to be KR only.**
- **ixgbe/base: Fix X550em link setup without SFP.**
- **ixgbe/base: Fix X550em mux after MAC reset.**

  Fixes: d2e72774e58c ("ixgbe/base: support X550")

- **ixgbe/base: Fix bus type overwrite.**
- **ixgbe/base: Fix init handling of X550em link down.**
- **ixgbe/base: Fix lan id before first i2c access.**
- **ixgbe/base: Fix mac type checks.**

- **ixgbe/base: Fix tunneled UDP and TCP frames in flow director.**
- **ixgbe: Check mbuf refcnt when clearing a ring.**

  The function to clear the TX ring when a port was being closed, e.g. on exit in testpmd, was not checking the mbuf refcnt before freeing it. Since the function in the vector driver to clear the ring after TX does not setting the pointer to NULL post-free, this caused crashes if mbuf debugging was turned on.
- **ixgbe: Fix RX with buffer address not word aligned.**

  Niantic HW expects the Header Buffer Address in the RXD must be word aligned.
- **ixgbe: Fix RX with buffer address not word aligned.**
- **ixgbe: Fix Rx queue reset.**

  Fix to reset vector related RX queue fields to their initial values.

  Fixes: c95584dc2b18 (“ixgbe: new vectorized functions for Rx/Tx”)
- **ixgbe: Fix TSO in IPv6.**

  When TSO was used with IPv6, the generated frames were incorrect. The L4 frame was OK, but the length field of IPv6 header was not populated correctly.
- **ixgbe: Fix X550 flow director check.**
- **ixgbe: Fix check for split packets.**

  The check for split packets to be reassembled in the vector ixgbe PMD was incorrectly only checking the first 16 elements of the array instead of all 32.

  Fixes: cf4b4708a88a (“ixgbe: improve slow-path perf with vector scattered Rx”)
- **ixgbe: Fix data access on big endian cpu.**
- **ixgbe: Fix flow director flexbytes offset.**

  Fixes: d54a9888267c (“ixgbe: support flexpayload configuration of flow director”)
- **ixgbe: Fix number of segments with vector scattered Rx.**

  Fixes: cf4b4708a88a (ixgbe: improve slow-path perf with vector scattered Rx)
- **ixgbe: Fix offload config option name.**

  The RX_OLFLAGS option was renamed from DISABLE to ENABLE in the driver code and Linux config. It is now renamed also in the BSD config and documentation.

  Fixes: 359f106a69a9 (“ixgbe: prefer enabling olflags rather than not disabling”)
- **ixgbe: Fix release queue mbufs.**

  The calculations of what mbufs were valid in the RX and TX queues were incorrect when freeing the mbufs for the vector PMD. This led to crashes due to invalid reference counts when mbuf debugging was turned on, and possibly other more subtle problems (such as mbufs being freed when in use) in other cases.

  Fixes: c95584dc2b18 (“ixgbe: new vectorized functions for Rx/Tx”)
- **ixgbe: Move PMD specific fields out of base driver.**

  Move `rx_bulk_alloc_allowed` and `rx_vec_allowed` from `ixgbe_hw` to `ixgbe_adapter`.

Fixes: 01fa1d6215fa ("ixgbe: unify Rx setup")

- **ixgbe: Rename TX queue release function.**
- **ixgbevf: Fix RX function selection.**

  The logic to select ixgbe the VF RX function is different than the PF.

- **ixgbevf: Fix link status for PF up/down events.**
- **kni: Fix RX loop limit.**

  Loop processing packets dequeued from rx_q was using the number of packets requested, not how many it actually received.

- **kni: Fix ioctl in containers, like Docker.**
- **kni: Fix multicast ioctl handling.**
- **log: Fix crash after log_history dump.**
- **lpm: Fix big endian support.**
- **lpm: Fix depth small entry add.**
- **mbuf: Fix cloning with private mbuf data.**

  Added a new `priv_size` field in mbuf structure that should be initialized at mbuf pool creation. This field contains the size of the application private data in mbufs.

  Introduced new static inline functions `rte_mbuf_from_indirect()` and `rte_mbuf_to_baddr()` to replace the existing macros, which take the private size into account when attaching and detaching mbufs.

- **mbuf: Fix data room size calculation in pool init.**

  Deduct the mbuf data room size from `mempool->elt_size` and `priv_size`, instead of using an hardcoded value that is not related to the real buffer size.

  To use `rte_pktmbuf_pool_init()`, the user can either:

  – Give a NULL parameter to rte_pktmbuf_pool_init(): in this case, the private size is assumed to be 0, and the room size is `mp->elt_size - sizeof(struct rte_mbuf)`.

  – Give the `rte_pktmbuf_pool_private` filled with appropriate data_room_size and priv_size values.

- **mbuf: Fix init when private size is not zero.**

  Allow the user to use the default `rte_pktmbuf_init()` function even if the mbuf private size is not 0.

- **mempool: Add structure for object headers.**

  Each object stored in mempools are prefixed by a header, allowing for instance to retrieve the mempool pointer from the object. When debug is enabled, a cookie is also added in this header that helps to detect corruptions and double-frees.

  Introduced a structure that materializes the content of this header, and will simplify future patches adding things in this header.

- **mempool: Fix pages computation to determine number of objects.**

- **mempool: Fix returned value after counting objects.**

  Fixes: 148f963fb532 ("xen: core library changes")

- **mlx4: Avoid requesting TX completion events to improve performance.**

  Instead of requesting a completion event for each TX burst, request it on a fixed schedule once every MLX4_PMD_TX_PER_COMP_REQ (currently 64) packets to improve performance.

- **mlx4: Fix compilation as a shared library and on 32 bit platforms.**

- **mlx4: Fix possible crash on scattered mbuf allocation failure.**

  Fixes issue where failing to allocate a segment, `mlx4_rx_burst_sp()` could call `rte_pktmbuf_free()` on an incomplete scattered mbuf whose next pointer in the last segment is not set.

- **mlx4: Fix support for multiple vlan filters.**

  This fixes the "Multiple RX VLAN filters can be configured, but only the first one works" bug.

- **pcap: Fix storage of name and type in queues.**

  pcap_rx_queue/pcap_tx_queue should store it's own copy of name/type values, not the pointer to temporary allocated space.

- **pci: Fix memory leaks and needless increment of map address.**

- **pci: Fix uio mapping differences between linux and bsd.**

- **port: Fix unaligned access to metadata.**

  Fix RTE_MBUF_METADATA macros to allow for unaligned accesses to meta-data fields.

- **ring: Fix return of new port id on creation.**

- **timer: Fix race condition.**

  Eliminate problematic race condition in `rte_timer_manage()` that can lead to corruption of per-lcore pending-lists (implemented as skip-lists).

- **vfio: Fix overflow of BAR region offset and size.**

  Fixes: 90a1633b2347 ("eal/Linux: allow to map BARs with MSI-X tables")

- **vhost: Fix enqueue/dequeue to handle chained vring descriptors.**

- **vhost: Fix race for connection fd.**

- **vhost: Fix virtio freeze due to missed interrupt.**

- **virtio: Fix crash if CQ is not negotiated.**

  Fix NULL dereference if virtio control queue is not negotiated.

- **virtio: Fix ring size negotiation.**

  Negotiate the virtio ring size. The host may allow for very large rings but application may only want a smaller ring. Conversely, if the number of descriptors requested exceeds the virtio host queue size, then just silently use the smaller host size.

  This fixes issues with virtio in non-QEMU environments. For example Google Compute Engine allows up to 16K elements in ring.

- **vmxnet3: Fix link state handling.**

13.3 Known Issues

- When running the `vmdq` sample or `vhost` sample applications with the Intel(R) XL710 (i40e) NIC, the configuration option `CONFIG_RTE_MAX_QUEUES_PER_PORT` should be increased from 256 to 1024.
- VM power manager may not work on systems with more than 64 cores.

13.4 API Changes

- The order that user supplied RX and TX callbacks are called in has been changed to the order that they were added (fifo) in line with end-user expectations. The previous calling order was the reverse of this (lifo) and was counter intuitive for users. The actual API is unchanged.

13.5 ABI Changes

- The `rte_hash` structure has been changed to internal use only.

CHAPTER

FOURTEEN

DPDK RELEASE 2.0

14.1 New Features

- Poll-mode driver support for an early release of the PCIE host interface of the Intel(R) Ethernet Switch FM10000.
  - Basic Rx/Tx functions for PF/VF
  - Interrupt handling support for PF/VF
  - Per queue start/stop functions for PF/VF
  - Support Mailbox handling between PF/VF and PF/Switch Manager
  - Receive Side Scaling (RSS) for PF/VF
  - Scatter receive function for PF/VF
  - Reta update/query for PF/VF
  - VLAN filter set for PF
  - Link status query for PF/VF

Note: The software is intended to run on pre-release hardware and may contain unknown or unresolved defects or issues related to functionality and performance. The poll mode driver is also pre-release and will be updated to a released version post hardware and base driver release. Should the official hardware release be made between DPDK releases an updated poll-mode driver will be made available.

- Link Bonding
  - Support for adaptive load balancing (mode 6) to the link bonding library.
  - Support for registration of link status change callbacks with link bonding devices.
  - Support for slaves devices which do not support link status change interrupts in the link bonding library via a link status polling mechanism.
- PCI Hotplug with NULL PMD sample application
- ABI versioning
- x32 ABI
- Non-EAL Thread Support
- Multi-pthread Support

- Re-order Library
- ACL for AVX2
- Architecture Independent CRC Hash
- uio_pci_generic Support
- KNI Optimizations
- Vhost-user support
- Virtio (link, vlan, mac, port IO, perf)
- IXGBE-VF RSS
- RX/TX Callbacks
- Unified Flow Types
- Indirect Attached MBUF Flag
- Use default port configuration in TestPMD
- Tunnel offloading in TestPMD
- Poll Mode Driver - 40 GbE Controllers (librte_pmd_i40e)
  - Support for Flow Director
  - Support for ethertype filter
  - Support RSS in VF
  - Support configuring redirection table with different size from 1GbE and 10 GbE
  - 128/512 entries of 40GbE PF
  - 64 entries of 40GbE VF
  - Support configuring hash functions
  - Support for VXLAN packet on Intel® 40GbE Controllers
- Poll Mode Driver for Mellanox ConnectX-3 EN adapters (mlx4)

Note: This PMD is only available for Linux and is disabled by default due to external dependencies (libibverbs and libmlx4). Please refer to the NIC drivers guide for more information.

- Packet Distributor Sample Application
- Job Stats library and Sample Application.
- Enhanced Jenkins hash (jhash) library

Note: The hash values returned by the new jhash library are different from the ones returned by the previous library.

CHAPTER

FIFTEEN

DPDK RELEASE 1.8

15.1 New Features

- Link Bonding
  - Support for 802.3ad link aggregation (mode 4) and transmit load balancing (mode 5) to the link bonding library.
  - Support for registration of link status change callbacks with link bonding devices.
  - Support for slaves devices which do not support link status change interrupts in the link bonding library via a link status polling mechanism.
- Poll Mode Driver - 40 GbE Controllers (librte_pmd_i40e)
  - Support for Flow Director
  - Support for ethertype filter
  - Support RSS in VF
  - Support configuring redirection table with different size from 1GbE and 10 GbE
  - 128/512 entries of 40GbE PF
  - 64 entries of 40GbE VF
  - Support configuring hash functions
  - Support for VXLAN packet on Intel 40GbE Controllers
- Packet Distributor Sample Application

CHAPTER

SIXTEEN

KNOWN ISSUES AND LIMITATIONS IN LEGACY RELEASES

This section describes known issues with the DPDK software that aren't covered in the version specific release notes sections.

16.1 Unit Test for Link Bonding may fail at test_tlb_tx_burst()

Description: Unit tests will fail in `test_tlb_tx_burst()` function with error for uneven distribution of packets.

Implication: Unit test link_bonding_autotest will fail.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: Fedora 20.

Driver/Module: Link Bonding.

16.2 Pause Frame Forwarding does not work properly on igb

Description: For igb devices rte_eth_flow_ctrl_set does not work as expected. Pause frames are always forwarded on igb, regardless of the `RFCE`, `MPMCF` and `DPF` registers.

Implication: Pause frames will never be rejected by the host on 1G NICs and they will always be forwarded.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.3 In packets provided by the PMD, some flags are missing

Description: In packets provided by the PMD, some flags are missing. The application does not have access to information provided by the hardware (packet is broadcast, packet is multicast, packet is IPv4 and so on).

Implication: The `ol_flags` field in the `rte_mbuf` structure is not correct and should not be used.

Resolution/Workaround: The application has to parse the Ethernet header itself to get the information, which is slower.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.4 The rte_malloc library is not fully implemented

Description: The `rte_malloc` library is not fully implemented.

Implication: All debugging features of rte_malloc library described in architecture documentation are not yet implemented.

Resolution/Workaround: No workaround available.

Affected Environment/Platform: All.

Driver/Module: `rte_malloc`.

16.5 HPET reading is slow

Description: Reading the HPET chip is slow.

Implication: An application that calls `rte_get_hpet_cycles()` or `rte_timer_manage()` runs slower.

Resolution/Workaround: The application should not call these functions too often in the main loop. An alternative is to use the TSC register through `rte_rdtsc()` which is faster, but specific to an lcore and is a cycle reference, not a time reference.

Affected Environment/Platform: All.

Driver/Module: Environment Abstraction Layer (EAL).

16.6 HPET timers do not work on the Osage customer reference platform

Description: HPET timers do not work on the Osage customer reference platform which includes an Intel® Xeon® processor 5500 series processor) using the released BIOS from Intel.

Implication: On Osage boards, the implementation of the `rte_delay_us()` function must be changed to not use the HPET timer.

Resolution/Workaround: This can be addressed by building the system with the `CONFIG_RTE_LIBEAL_USE_HPET=n` configuration option or by using the `--no-hpet` EAL option.

Affected Environment/Platform: The Osage customer reference platform. Other vendor platforms with Intel® Xeon® processor 5500 series processors should work correctly, provided the BIOS supports HPET.

Driver/Module: `lib/librte_eal/common/include/rte_cycles.h`

16.7 Not all variants of supported NIC types have been used in testing

Description: The supported network interface cards can come in a number of variants with different device ID's. Not all of these variants have been tested with the DPDK.

The NIC device identifiers used during testing:

- Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1584]
- Intel® Ethernet Controller XL710 for 40GbE QSFP+ [8086:1583]
- Intel® Ethernet Controller X710 for 10GbE SFP+ [8086:1572]
- Intel® 82576 Gigabit Ethernet Controller [8086:10c9]
- Intel® 82576 Quad Copper Gigabit Ethernet Controller [8086:10e8]
- Intel® 82580 Dual Copper Gigabit Ethernet Controller [8086:150e]
- Intel® I350 Quad Copper Gigabit Ethernet Controller [8086:1521]
- Intel® 82599 Dual Fibre 10 Gigabit Ethernet Controller [8086:10fb]
- Intel® Ethernet Server Adapter X520-T2 [8086: 151c]
- Intel® Ethernet Controller X540-T2 [8086:1528]
- Intel® 82574L Gigabit Network Connection [8086:10d3]
- Emulated Intel® 82540EM Gigabit Ethernet Controller [8086:100e]
- Emulated Intel® 82545EM Gigabit Ethernet Controller [8086:100f]
- Intel® Ethernet Server Adapter X520-4 [8086:154a]
- Intel® Ethernet Controller I210 [8086:1533]

Implication: Risk of issues with untested variants.

Resolution/Workaround: Use tested NIC variants. For those supported Ethernet controllers, additional device IDs may be added to the software if required.

Affected Environment/Platform: All.

Driver/Module: Poll-mode drivers

16.8 Multi-process sample app requires exact memory mapping

Description: The multi-process example application assumes that it is possible to map the hugepage memory to the same virtual addresses in client and server applications. Occasionally, very rarely with 64-bit, this does not occur and a client application will fail on startup. The Linux "address-space layout randomization" security feature can sometimes cause this to occur.

Implication: A multi-process client application fails to initialize.

Resolution/Workaround: See the "Multi-process Limitations" section in the DPDK Programmer's Guide for more information.

Affected Environment/Platform: All.

Driver/Module: Multi-process example application

16.9 Packets are not sent by the 1 GbE/10 GbE SR-IOV driver when the source MAC is not the MAC assigned to the VF NIC

Description: The 1 GbE/10 GbE SR-IOV driver can only send packets when the Ethernet header's source MAC address is the same as that of the VF NIC. The reason for this is that the Linux `ixgbe` driver module in the host OS has its anti-spoofing feature enabled.

Implication: Packets sent using the 1 GbE/10 GbE SR-IOV driver must have the source MAC address correctly set to that of the VF NIC. Packets with other source address values are dropped by the NIC if the application attempts to transmit them.

Resolution/Workaround: Configure the Ethernet source address in each packet to match that of the VF NIC.

Affected Environment/Platform: All.

Driver/Module: 1 GbE/10 GbE VF Poll Mode Driver (PMD).

16.10 SR-IOV drivers do not fully implement the rte_ethdev API

Description: The SR-IOV drivers only supports the following rte_ethdev API functions:

- rte_eth_dev_configure()
- rte_eth_tx_queue_setup()
- rte_eth_rx_queue_setup()
- rte_eth_dev_info_get()
- rte_eth_dev_start()
- rte_eth_tx_burst()
- rte_eth_rx_burst()
- rte_eth_dev_stop()
- rte_eth_stats_get()
- rte_eth_stats_reset()
- rte_eth_link_get()
- rte_eth_link_get_no_wait()

Implication: Calling an unsupported function will result in an application error.

Resolution/Workaround: Do not use other rte_ethdev API functions in applications that use the SR-IOV drivers.

Affected Environment/Platform: All.

Driver/Module: VF Poll Mode Driver (PMD).

16.11 PMD does not work with –no-huge EAL command line parameter

Description: Currently, the DPDK does not store any information about memory allocated by `malloc()` (for example, NUMA node, physical address), hence PMD drivers do not work when the ``--no-huge` command line parameter is supplied to EAL.

Implication: Sending and receiving data with PMD will not work.

Resolution/Workaround: Use huge page memory or use VFIO to map devices.

Affected Environment/Platform: Systems running the DPDK on Linux

Driver/Module: Poll Mode Driver (PMD).

16.12 Some hardware off-load functions are not supported by the VF Driver

Description: Currently, configuration of the following items is not supported by the VF driver:

- IP/UDP/TCP checksum offload
- Jumbo Frame Receipt
- HW Strip CRC

Implication: Any configuration for these items in the VF register will be ignored. The behavior is dependent on the current PF setting.

Resolution/Workaround: For the PF (Physical Function) status on which the VF driver depends, there is an option item under PMD in the config file. For others, the VF will keep the same behavior as PF setting.

Affected Environment/Platform: All.

Driver/Module: VF (SR-IOV) Poll Mode Driver (PMD).

16.13 Kernel crash on IGB port unbinding

Description: Kernel crash may occur when unbinding 1G ports from the igb_uio driver, on 2.6.3x kernels such as shipped with Fedora 14.

Implication: Kernel crash occurs.

Resolution/Workaround: Use newer kernels or do not unbind ports.

Affected Environment/Platform: 2.6.3x kernels such as shipped with Fedora 14

Driver/Module: IGB Poll Mode Driver (PMD).

16.14 Twinpond and Ironpond NICs do not report link status correctly

Description: Twin Pond/Iron Pond NICs do not bring the physical link down when shutting down the port.

Implication: The link is reported as up even after issuing `shutdown` command unless the cable is physically disconnected.

Resolution/Workaround: None.

Affected Environment/Platform: Twin Pond and Iron Pond NICs

Driver/Module: Poll Mode Driver (PMD).

16.15 Discrepancies between statistics reported by different NICs

Description: Gigabit Ethernet devices from Intel include CRC bytes when calculating packet reception statistics regardless of hardware CRC stripping state, while 10-Gigabit Ethernet devices from Intel do so only when hardware CRC stripping is disabled.

Implication: There may be a discrepancy in how different NICs display packet reception statistics.

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.16 Error reported opening files on DPDK initialization

Description: On DPDK application startup, errors may be reported when opening files as part of the initialization process. This occurs if a large number, for example, 500 or more, or if hugepages are used, due to the per-process limit on the number of open files.

Implication: The DPDK application may fail to run.

Resolution/Workaround: If using 2 MB hugepages, consider switching to a fewer number of 1 GB pages. Alternatively, use the `ulimit` command to increase the number of files which can be opened by a process.

Affected Environment/Platform: All.

Driver/Module: Environment Abstraction Layer (EAL).

16.17 Intel® QuickAssist Technology sample application does not work on a 32-bit OS on Shumway

Description: The Intel® Communications Chipset 89xx Series device does not fully support NUMA on a 32-bit OS. Consequently, the sample application cannot work properly on Shumway, since it requires NUMA on both nodes.

Implication: The sample application cannot work in 32-bit mode with emulated NUMA, on multi-socket boards.

Resolution/Workaround: There is no workaround available.

Affected Environment/Platform: Shumway

Driver/Module: All.

16.18 Differences in how different Intel NICs handle maximum packet length for jumbo frame

Description: 10 Gigabit Ethernet devices from Intel do not take VLAN tags into account when calculating packet size while Gigabit Ethernet devices do so for jumbo frames.

Implication: When receiving packets with VLAN tags, the actual maximum size of useful payload that Intel Gigabit Ethernet devices are able to receive is 4 bytes (or 8 bytes in the case of packets with extended VLAN tags) less than that of Intel 10 Gigabit Ethernet devices.

Resolution/Workaround: Increase the configured maximum packet size when using Intel Gigabit Ethernet devices.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.19 Binding PCI devices to igb_uio fails on Linux kernel 3.9 when more than one device is used

Description: A known bug in the uio driver included in Linux kernel version 3.9 prevents more than one PCI device to be bound to the igb_uio driver.

Implication: The Poll Mode Driver (PMD) will crash on initialization.

Resolution/Workaround: Use earlier or later kernel versions, or apply the following patch.

Affected Environment/Platform: Linux systems with kernel version 3.9

Driver/Module: igb_uio module

16.20 GCC might generate Intel® AVX instructions for processors without Intel® AVX support

Description: When compiling DPDK (and any DPDK app), gcc may generate Intel® AVX instructions, even when the processor does not support Intel® AVX.

Implication: Any DPDK app might crash while starting up.

Resolution/Workaround: Either compile using icc or set `EXTRA_CFLAGS='-O3'` prior to compilation.

Affected Environment/Platform: Platforms which processor does not support Intel® AVX.

Driver/Module: Environment Abstraction Layer (EAL).

16.21 Ethertype filter could receive other packets (non-assigned) in Niantic

Description: On Intel® Ethernet Controller 82599EB When Ethertype filter (priority enable) was set, unmatched packets also could be received on the assigned queue, such as ARP packets without 802.1q tags or with the user priority not equal to set value. Launch the testpmd by disabling RSS and with multiply queues, then add the ethertype filter like the following and then start forwarding:

```
add_ethertype_filter 0 ethertype 0x0806 priority enable 3 queue 2 index 1
```

When sending ARP packets without 802.1q tag and with user priority as non-3 by tester, all the ARP packets can be received on the assigned queue.

Implication: The user priority comparing in Ethertype filter cannot work probably. It is a NIC's issue due to the following: "In fact, ETQF.UP is not functional, and the information will be added in errata of 82599 and X540."

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.22 Cannot set link speed on Intel® 40G Ethernet controller

Description: On Intel® 40G Ethernet Controller you cannot set the link to specific speed.

Implication: The link speed cannot be changed forcibly, though it can be configured by application.

Resolution/Workaround: None

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.23 Devices bound to igb_uio with VT-d enabled do not work on Linux kernel 3.15-3.17

Description: When VT-d is enabled (`iommu=pt intel_iommu=on`), devices are 1:1 mapped. In the Linux kernel unbinding devices from drivers removes that mapping which result in IOMMU errors. Introduced in Linux kernel 3.15 commit, solved in Linux kernel 3.18 commit.

Implication: Devices will not be allowed to access memory, resulting in following kernel errors:

```
dmar: DRHD: handling fault status reg 2
dmar: DMAR:[DMA Read] Request device [02:00.0] fault addr a0c58000
DMAR:[fault reason 02] Present bit in context entry is clear
```

Resolution/Workaround: Use earlier or later kernel versions, or avoid driver binding on boot by blacklisting the driver modules. I.e., in the case of `ixgbe`, we can pass the kernel command line option: `modprobe.blacklist=ixgbe`. This way we do not need to unbind the device to bind it to igb_uio.

Affected Environment/Platform: Linux systems with kernel versions 3.15 to 3.17.

Driver/Module: `igb_uio` module.

16.24 VM power manager may not work on systems with more than 64 cores

Description: When using VM power manager on a system with more than 64 cores, VM(s) should not use cores 64 or higher.

Implication: VM power manager should not be used with VM(s) that are using cores 64 or above.

Resolution/Workaround: Do not use cores 64 or above.

Affected Environment/Platform: Platforms with more than 64 cores.

Driver/Module: VM power manager application.

16.25 DPDK may not build on some Intel CPUs using clang < 3.7.0

Description: When compiling DPDK with an earlier version than 3.7.0 of clang, CPU flags are not detected on some Intel platforms such as Intel Broadwell/Skylake (and possibly future CPUs), and therefore compilation fails due to missing intrinsics.

Implication: DPDK will not build when using a clang version < 3.7.0.

Resolution/Workaround: Use clang 3.7.0 or higher, or gcc.

Affected Environment/Platform: Platforms with Intel Broadwell/Skylake using an old clang version.

Driver/Module: Environment Abstraction Layer (EAL).

16.26 The last EAL argument is replaced by the program name in argv[]

Description: The last EAL argument is replaced by program name in `argv[]` after `eal_parse_args` is called. This is the intended behavior but it causes the pointer to the last EAL argument to be lost.

Implication: If the last EAL argument in `argv[]` is generated by a malloc function, changing it will cause memory issues when freeing the argument.

Resolution/Workaround: An application should not consider the value in `argv[]` as unchanged.

Affected Environment/Platform: ALL.

Driver/Module: Environment Abstraction Layer (EAL).

16.27 I40e VF may not receive packets in the promiscuous mode

Description: Promiscuous mode is not supported by the DPDK i40e VF driver when using the i40e Linux kernel driver as host driver.

Implication: The i40e VF does not receive packets when the destination MAC address is unknown.

Resolution/Workaround: Use a explicit destination MAC address that matches the VF.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.28 uio pci generic module bind failed in X710/XL710/XXV710

Description: The `uio_pci_generic` module is not supported by XL710, since the errata of XL710 states that the Interrupt Status bit is not implemented. The errata is the item #71 from the xl710 controller spec. The hw limitation is the same as other X710/XXV710 NICs.

Implication: When use `--bind=uio_pci_generic`, the `uio_pci_generic` module probes device and check the Interrupt Status bit. Since it is not supported by X710/XL710/XXV710, it return a *failed* value. The statement that these products don't support INTx masking, is indicated in the related linux kernel commit.

Resolution/Workaround: Do not bind the `uio_pci_generic` module in X710/XL710/XXV710 NICs.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.29 virtio tx_burst() function cannot do TSO on shared packets

Description: The standard TX function of virtio driver does not manage shared packets properly when doing TSO. These packets should be read-only but the driver modifies them.

When doing TSO, the virtio standard expects that the L4 checksum is set to the pseudo header checksum in the packet data, which is different than the DPDK API. The driver patches the L4 checksum to conform to the virtio standard, but this solution is invalid when dealing with shared packets (clones), because the packet data should not be modified.

Implication: In this situation, the shared data will be modified by the driver, potentially causing race conditions with the other users of the mbuf data.

Resolution/Workaround: The workaround in the application is to ensure that the network headers in the packet data are not shared.

Affected Environment/Platform: Virtual machines running a virtio driver.

Driver/Module: Poll Mode Driver (PMD).

16.30 igb uio legacy mode can not be used in X710/XL710/XXV710

Description: X710/XL710/XXV710 NICs lack support for indicating INTx is asserted via the interrupt bit in the PCI status register. Linux delected them from INTx support table. The related commit.

Implication: When insmod `igb_uio` with `intr_mode=legacy` and test link status interrupt. Since INTx interrupt is not supported by X710/XL710/XXV710, it will cause Input/Output error when reading file descriptor.

Resolution/Workaround: Do not bind `igb_uio` with legacy mode in X710/XL710/XXV710 NICs, or do not use kernel version >4.7 when you bind `igb_uio` with legacy mode.

Affected Environment/Platform: ALL.

Driver/Module: Poll Mode Driver (PMD).

16.31 igb_uio can not be used when running l3fwd-power

Description: Link Status Change(LSC) interrupt and packet receiving interrupt are all enabled in l3fwd-power APP. Because of UIO only support one interrupt, so these two kinds of interrupt need to share one, and the receiving interrupt have the higher priority, so can't get the right link status.

Implication: When insmod `igb_uio` and running l3fwd-power APP, link status getting doesn't work properly.

Resolution/Workaround: Use vfio-pci when LSC and packet receiving interrupt enabled.

Affected Environment/Platform: ALL.

Driver/Module: `igb_uio` module.

16.32 Linux kernel 4.10.0 iommu attribute read error

Description: When VT-d is enabled (`iommu=pt intel_iommu=on`), reading IOMMU attributes from /sys/devices/virtual/iommu/dmarXXX/intel-iommu/cap on Linux kernel 4.10.0 error. This bug is fixed in Linux commmit a7fdb6e648fb, This bug is introduced in Linux commmit 39ab9555c241,

Implication: When binding devices to VFIO and attempting to run testpmd application, testpmd (and other DPDK applications) will not initialize.

Resolution/Workaround: Use other linux kernel version. It only happens in linux kernel 4.10.0.

Affected Environment/Platform: ALL OS of linux kernel 4.10.0.

Driver/Module: `vfio-pci` module.

16.33 Netvsc driver and application restart

Description: The Linux kernel uio_hv_generic driver does not completely shutdown and clean up resources properly if application using Netvsc PMD exits.

Implication: When application using Netvsc PMD is restarted it can not complete initialization handshake sequence with the host.

Resolution/Workaround: Either reboot the guest or remove and reinsert the hv_uio_generic module.

Affected Environment/Platform: Linux Hyper-V.

Driver/Module: `uio_hv_generic` module.

16.34 PHY link up fails when rebinding i40e NICs to kernel driver

Description: Some kernel drivers are not able to handle the link status correctly after DPDK application sets the PHY to link down.

Implication: The link status can't be set to "up" after the NIC is rebound to the kernel driver. Before a DPDK application quits it will invoke the function `i40e_dev_stop()` which will sets the PHY to link down. Some kernel drivers may not be able to handle the link status correctly after it retakes control of the device. This is a known PHY link configuration issue in the i40e kernel driver. The fix has been addressed in the 2.7.4 rc version. So if the i40e kernel driver is < 2.7.4 and doesn't have the fix backported it will encounter this issue.

Resolution/Workaround: First try to remove and reinsert the i40e kernel driver. If that fails reboot the system.

Affected Environment/Platform: All.

Driver/Module: Poll Mode Driver (PMD).

16.35 Restricted vdev ethdev operations supported in secondary process

Description In current virtual device sharing model, Ethernet device data structure will be shared between primary and secondary process. Only those Ethernet device operations which based on it are workable in secondary process.

Implication Some Ethernet device operations like device start/stop will be failed on virtual device in secondary process.

Affected Environment/Platform: ALL.

Driver/Module: Virtual Device Poll Mode Driver (PMD).

16.36 Kernel crash when hot-unplug igb_uio device while DPDK application is running

Description: When device has been bound to igb_uio driver and application is running, hot-unplugging the device may cause kernel crash.

Reason: When device is hot-unplugged, igb_uio driver will be removed which will destroy UIO resources. Later trying to access any uio resource will cause kernel crash.

Resolution/Workaround: If using DPDK for PCI HW hot-unplug, prefer to bind device with VFIO instead of IGB_UIO.

Affected Environment/Platform: ALL.

Driver/Module: `igb_uio` module.

CHAPTER

SEVENTEEN

ABI AND API DEPRECATION

See the `guidelines document for details of the ABI policy`. API and ABI deprecation notices are to be posted here.

17.1 Deprecation Notices

- linux: Linux kernel version 3.2 (which is the current minimum required version for the DPDK) is not maintained anymore. Therefore the planned minimum required kernel version for DPDK 19.02 will be the next oldest Long Term Stable (LTS) version which is 3.16, but compatibility for recent distribution kernels will be kept.
- kvargs: The function `rte_kvargs_process` will get a new parameter for returning key match count. It will ease handling of no-match case.
- eal: function `rte_bsf64` in `rte_bitmap.h` has been renamed to `rte_bsf64_safe` and moved to `rte_common.h`. A new `rte_bsf64` function will be added in the next release in `rte_common.h` that follows convention set by existing `rte_bsf32` function.
- eal: both declaring and identifying devices will be streamlined in v18.11. New functions will appear to query a specific port from buses, classes of device and device drivers. Device declaration will be made coherent with the new scheme of device identification. As such, `rte_devargs` device representation will change.
  - The enum `rte_devtype` was used to identify a bus and will disappear.
  - Functions previously deprecated will change or disappear:
    * `rte_eal_devargs_type_count`
- pci: Several exposed functions are misnamed. The following functions are deprecated starting from v17.11 and are replaced:
  - `eal_parse_pci_BDF` replaced by `rte_pci_addr_parse`
  - `eal_parse_pci_DomBDF` replaced by `rte_pci_addr_parse`
  - `rte_eal_compare_pci_addr` replaced by `rte_pci_addr_cmp`
- dpaa2: removal of `rte_dpaa2_memsegs` structure which has been replaced by a pa-va search library. This structure was earlier being used for holding memory segments used by dpaa2 driver for faster pa->va translation. This structure would be made internal (or removed if all dependencies are cleared) in future releases.

- mbuf: The opaque `mbuf->hash.sched` field will be updated to support generic definition in line with the ethdev TM and MTR APIs. Currently, this field is defined in librte_sched in a non-generic way. The new generic format will contain: queue ID, traffic class, color. Field size will not change.
- sched: Some API functions will change prototype due to the above deprecation note for mbuf->hash.sched, e.g. `rte_sched_port_pkt_write()` and `rte_sched_port_pkt_read()` will likely have an additional parameter of type `struct rte_sched_port`.
- mbuf: the macro `RTE_MBUF_INDIRECT()` will be removed in v18.08 or later and replaced with `RTE_MBUF_CLONED()` which is already added in v18.05. As `EXT_ATTACHED_MBUF` is newly introduced in v18.05, `RTE_MBUF_INDIRECT()` can no longer be mutually exclusive with `RTE_MBUF_DIRECT()` if the new experimental API `rte_pktmbuf_attach_extbuf()` is used. Removal of the macro is to fix this semantic inconsistency.
- ethdev: the legacy filter API, including `rte_eth_dev_filter_supported()`, `rte_eth_dev_filter_ctrl()` as well as filter types MACVLAN, ETHERTYPE, FLEXIBLE, SYN, NTUPLE, TUNNEL, FDIR, HASH and L2_TUNNEL, is superseded by the generic flow API (rte_flow) in PMDs that implement the latter. Target release for removal of the legacy API will be defined once most PMDs have switched to rte_flow.
- ethdev: Maximum and minimum MTU values vary between hardware devices. In hardware agnostic DPDK applications access to such information would allow a more accurate way of validating and setting supported MTU values on a per device basis rather than using a defined default for all devices. To resolve this, the following members will be added to `rte_eth_dev_info`. Note: these can be added to fit a hole in the existing structure for amd64 but not for 32-bit, as such ABI change will occur as size of the structure will increase.
  - Member `uint16_t min_mtu` the minimum MTU allowed.
  - Member `uint16_t max_mtu` the maximum MTU allowed.
- security: New field `uint64_t opaque_data` is planned to be added into `rte_security_session` structure. That would allow upper layer to easily associate/de-associate some user defined data with the security session.
- cryptodev: several API and ABI changes are planned for rte_cryptodev in v19.02:
  - The size and layout of `rte_cryptodev_sym_session` will change to fix existing issues.
  - The size and layout of `rte_cryptodev_qp_conf` and syntax of `rte_cryptodev_queue_pair_setup` will change to to allow to use two different mempools for crypto and device private sessions.
- pdump: As we changed to use generic IPC, some changes in APIs and structure are expected in subsequent release.
  - `rte_pdump_set_socket_dir` will be removed;
  - The parameter, `path`, of `rte_pdump_init` will be removed;
  - The enum `rte_pdump_socktype` will be removed.