Using the mlx5 Tx datapath tracing feature

The mlx5 Tx datapath tracing feature provides the capability to gather the comprehensive information about packets handling in PMD with timings, including the packet sending completion ones.

1. Build DPDK application with enabled datapath tracing

The meson option should be specified:

```
--enable_trace_fp=true
```

The c_args shoudl be specified:

```
-DALLOW_EXPERIMENTAL_API
```

The DPDK configuration examples:

```
meson configure --buildtype=debug -Denable_trace_fp=true -Dc_args='-DRTE_LIBRTE_MLX5_DEBUG -DRTE_ENABLE_ASSERT -DALLOW_EXPERIMENTAL_API' build
```

```
meson configure --buildtype=debug -Denable_trace_fp=true -Dc_args='-DRTE_ENABLE_ASSERT -DALLOW_EXPERIMENTAL_API' build
```

```
meson configure --buildtype=release -Denable_trace_fp=true -Dc_args='-DRTE_ENABLE_ASSERT -DALLOW_EXPERIMENTAL_API' build
```

```
meson configure --buildtype=release -Denable_trace_fp=true -Dc_args='-DALLOW_EXPERIMENTAL_API' build
```

2. Configuring the NIC

If the sending completion timings are crucial the NIC should be configured to provide real-time timestamps, the REAL_TIME_CLOCK_ENABLE NV settings parameter should be configured to TRUE, for example with command (followed by FW/driver reset):

```
sudo mlxconfig -d /dev/mst/mt4125_pciconf0 s REAL_TIME_CLOCK_ENABLE=1
```

3. Run DPDK application to gather the traces

EAL parameters controlling trace capability in runtime:

`--trace=pmd.net.mlx5.tx`
the regular expression enabling the trace points with matching names at least "pmd.net.mlx5.tx" must be enabled to gather all events needed to analyze mlx5 Tx datapath and its timings. By default, all the trace points are disabled

`--trace-dir=/var/log`
- trace storing directory, optional, default is “$HOME/dpdk-traces/rte-yyyy-mm-dd-[AP]M-hh-mm-ss/”

`--trace-bufsz=<val>B|<val>K|<val>M`
optional, trace data buffer size per thread, optional, default is 1MB

`--trace-mode=overwrite|discard`
optional, selects trace data buffer mode

The trace data are committed to the specified folder on EAL cleanup. Optionally data commit can be triggered explicitly by application via `rte_trace_save()` API call.

4. Installing or building Babeltrace2 package

The gathered trace data can be analyzed with Python script.
To parse the trace data script uses Babeltrace2 library, the package should be either installed or built from source code:

```
git clone https://github.com/efficios/babeltrace.git
cd babeltrace
./bootstrap
./configure –help
./configure --disable-api-doc --disable-man-pages --disable-python-bindings-doc --enbale-python-plugins --enable-python-binding
```

5. Running the analyzing script

The analyzing script is located in the folder: ./drivers/net/mlx5/tools
It requires the Python3.6, Babeltrace2 packages and takes the only parameter of trace data folder, example:

```
./mlx5_trace.py /var/log/rte-2023-01-23-AM-11-52-39
```

6. Interpreting the script output data.

All timings are given in nanoseconds.
The list of Tx (and coming Rx) bursts per port/queue is presented in output. Each list element contains the list of build WQEs, and each WQE contains the list of packets to send.


Burst type, port, queue
Packets requested/sent
Tx_burst entry UTC
Tx_burst completed in ns
WQE opcode
WQE index
WQE completion after scheduled time in ns
WQE scheduled time or pushed time UTC
1674467558836934974: tx(p=0, q=0, 45/64 pkts in 5016
5931: WAIT (1674467556276837000, 180)
5932: EMPW (1674467556276837000, 392)
20B89FAC0: 100
20B89F180: 100
20B89E840: 100
20B89DF00: 100
20B89D5C0: 100
20B89CC80: 100
20B89C340: 100
20B89BA00: 100
5941: EMPW (1674467556276837000, 584)
20B89B0C0: 100
20B89A780: 100
20B899E40: 100
20B899500: 100
20B898BC0: 100
20B898280: 100
20B897940: 100
20B897000: 100
5950: EMPW (1674467556276837000, 680)
20B8966C0: 100
20B895D80: 100
20B895440: 100
20B894B00: 100
5958: EMPW (1674467556276837000, 876)
20B8941C0: 100
20B893880: 100
20B892F40: 100
20B892600: 100
20B891CC0: 100
20B891380: 100
20B890A40: 100
20B890100: 100
5967: EMPW (1674467556276837000, 944)
20B88F7C0: 100
20B88EE80: 100
20B88E540: 100