

vHost devices executed tests

Test Environment details:

- Bare Metal, Ubuntu 16.04.1 LTS

Repo/branch used:

- collectd/master branch (git clone [git://github.com/collectd/collectd.git](https://github.com/collectd/collectd.git)).

DPDK stats: DPDK vhost interface to retrieve stats.

Tests precondition:

- Latest DPDK installed (17.02, also try with DPDK 16.04).

DPDK configuration:

- [DPDK installation details](#).



- Configuration details -

Collectd configuration:

```
LoadPlugin dpdkstat
<Plugin dpdkstat>
  <EAL>
    Coremask "0x2"
    MemoryChannels "4"
    ProcessType "secondary"
    FilePrefix "rte"
  </EAL>
# SharedMemObj "dpdk_collectd_stats_0"
  EnabledPortMask 0xffff
# PortName "interface1"
# PortName "interface2"
</Plugin>
```

Table#1: DPDK stats vhost interface tests.

#	Test summary	Steps	Expected	Test Result	Comments
1	Verify vhost DPDK stats upon collectd restart.	1. Start collectd. Start testpmd. 2. Stop collectd. 3. Start collectd.	1. DPDK stats are updated. 2. DPDK stats are not updated. 3. DPDK stats are updated.		
2	Verify vhost DPDK stats with different time interval.	1. Change time interval to 1 second. Start collectd, testpmd. 2. Change time interval in range 2-90 seconds. Restart collectd.	1. DPDK stats are updated every second. 2. DPDK stats are updated every new defined time interval.		

3	Verify vhost DPDK stats with default configuration.	<ol style="list-style-type: none"> 1. Comment out "<Plugin dpdkstat>" section in collectd.conf. Start collectd, testpmd. 2. Define "collectd configuration" in collectd.conf file. Restart collectd service. 	<ol style="list-style-type: none"> 1. Collectd started with default configuration, with warnings? DPDK stats are updated every time interval. 2. Collectd started with given configuration. DPDK stats are updated every time interval. 		
4	Verify vhost DPDK stats counters number for available driver.	<ol style="list-style-type: none"> 1. Start collectd, testpmd. 2. Get all available vhost DPDK stats counters. 3. Provide list of counters available and driver. 	<ol style="list-style-type: none"> 1. Collectd started without errors. 2. Record number of counters. 3. Number and counters available. 		<p>MT: Please note the number of stats supported for a vhost port - will be only a subset of that of a physical port.</p> <p>SA: Just will dump counters for driver tested.</p>
5	Verify vhost DPDK Xstats counters number for available driver.	<ol style="list-style-type: none"> 1. Start collectd, testpmd. 2. Get all available vhost DPDK Xstats counters. 3. Compare them with DPDK Xstats counters. 	<ol style="list-style-type: none"> 1. Collectd started without errors. 2. Record number of counters. 3. Number and counters names are the same? 		
6	Verify vhost DPDK stats for Rx packets /bytes counters.	<ol style="list-style-type: none"> 1. Start collectd, testpmd. 2. Send traffic toward vhost port. 	<ol style="list-style-type: none"> 1. Collectd started without errors. 2. Vhost DPDK statistic for Rx packets bytes is same as sent, shown by collectd plugin. 		
7	Verify vhost DPDK stats for packets with different size.	<ol style="list-style-type: none"> 1. Start collectd, testpmd. 2. Send packets toward vhost with different packet size [bytes]: 64, 128, 256, 512, 1024, 1518. 3. Try to send packets with size [bytes]: 60, 2048, 10000. 	<ol style="list-style-type: none"> 1. Collectd started. 2. Vhost DPDK statistic for Rx packets sizes is same as sent, shown by collectd plugin. 3. Vhost DPDK statistic for Rx packets sizes is same as sent, shown by collectd plugin. 		
8	Verify vhost DPDK stats for unicast /multicast/broadcast packets.	<ol style="list-style-type: none"> 1. Start collectd, testpmd. 2. Send packets toward vhost: unicast, multicast, broadcast. 	<ol style="list-style-type: none"> 1. Collectd started. 2. Vhost DPDK statistic for unicast /multicast/broadcast is same as sent, shown by collectd plugin. 		
9	Verify vhost DPDK Xstats not available for DPDK 16.04.	<ol style="list-style-type: none"> 1. Install separately DPDK 16.04. 2. Reconfigure system to work with DPDK 16.04. 3. Get vhost DPDK stats. 	<ol style="list-style-type: none"> 1. Success. 2. Success (hopefully). 3. Xstats counters are not available for vhost DPDK. 		

DPDK events: DPDK vhost interface to retrieve link status.

Collectd configuration:

```

<Plugin "dpdkevents">
  <EAL>
    Coremask "0xf"
    MemoryChannels "4"
    ProcessType "secondary"
    FilePrefix "rte"
  </EAL>
  <Event "link_status">
    SendEventsOnUpdate true
    EnabledPortMask 0xffff
    SendNotification false
  </Event>
</Plugin>

```

Table#2: DPDK events vhost interface tests.

#	Test summary	Steps	Expected	Test Result	Comments
---	--------------	-------	----------	-------------	----------

1	Verify vhost DPDK metrics upon SendEventsOnUpdate set to true/false.	<ol style="list-style-type: none"> 1. Set "SendEventsOnUpdate true" in collectd.conf. 2. Start collectd, testpmd. 3. Set port up/down/up. 4. Set "SendEventsOnUpdate false" in collectd.conf. Restart collectd. 5. Set port up/down. 	<p>3. Verify link status is dispatched as statistic accordingly.</p> <p>5. Verify link status is not dispatched as statistic.</p>		
2	Verify vhost DPDK metrics are not collected for ports disabled by EnabledPortMask.	<ol style="list-style-type: none"> 1. Set "EnabledPortMask 0xFE". This will disable port 0. 2. Start collectd. Start testpmd. 3. Set port up/down and wait read interval. 4. Set "EnabledPortMask field to 0x01". Restart collectd. 5. Set port up/down and wait read interval. 	<p>3. Verify that link status is not dispatched for port this port.</p> <p>5. Verify link status is dispatched for this port.</p>		
3	Verify vhost DPDK metrics not collected for ports disabled by EnabledPortMask and disabled by SendEventsOnUpdate.	<ol style="list-style-type: none"> 1. Set "EnabledPortMask 0xFE", "SendEventsOnUpdate false". 2. Start collectd. Start testpmd. 3. Set port up/down and wait read interval. 	<p>3. Verify link status is not dispatched as statistic accordingly.</p>		
4	Verify link status is collected according to defined PortName field.	<ol style="list-style-type: none"> 1. Set "PortName 'test_port' " value. 2. Start collectd. Start testpmd. 3. Set port up/down/up. 	<p>3. Verify link status is dispatched for port name 'test_port'.</p>		
5	Verify vhost DPDK link status correct notification is sent when SendNotification and SendEventsOnUpdate are true.	<ol style="list-style-type: none"> 1. Set "SendEventsOnUpdate true", "SendNotification true". 2. Start collectd. Start testpmd. 3. Set port up/down/up. 	<p>3. Verify link status is dispatched only when link state is changed.</p> <p>Verify notification about link status changed is present in syslog.</p>		
6	Verify vhost DPDK link status correct notification is sent when SendNotification is true and SendEventsOnUpdate is false.	<ol style="list-style-type: none"> 1. Set "SendEventsOnUpdate false", "SendNotification true". 2. Start collectd. Start testpmd. 3. Set port up/down/up. 	<p>3. Verify link status is dispatched every read interval.</p> <p>Verify notification about link status changed is present in syslog.</p>		
7	Verify vhost DPDK link status is collected when default dpdkevents configuration is applied.	<ol style="list-style-type: none"> 1. In collectd.conf leave only "LoadPlugin "dpdkevents" line. 2. Start collectd. Start testpmd. 	<p>2. Verify link status is dispatched as statistic every read interval.</p>		
8	Verify vhost DPDK events plugin attempts to reconnect to primary process every read interval.	<ol style="list-style-type: none"> 1. Start collectd. Start testpmd. 2. Wait for at least one read interval. 3. Stop testpmd. 	<p>3. Check syslog for attempts to reconnect to primary process every read interval.</p>		

9	Verify vhost DPDK events plugin dispatch values every 5ms.	<ol style="list-style-type: none"> 1. Set read interval for dpdkevents plugin to 0.005 seconds. 2. Start testpmd. Start collectd. 3. Wait for at least one read interval. 	3. Verify values are dispatched every 5 ms.		
10	Verify vhost DPDK events plugin stops collecting link status data after primary process is stopped and resumes collecting when primary process is started again.	<ol style="list-style-type: none"> 1. Start collectd. Start testpmd. 2. Wait for at least one read interval. 3. Stop testpmd. 4. Start testpmd. 	<ol style="list-style-type: none"> 3. Wait read interval and check that no data is dispatched. 4. Wait read interval and check that data is dispatched. 		

OVS stats: native vhost interface (tap device) and DPDK vhost interface to retrieve statistic.

OVS stats: DPDK vhost interface to retrieve statistic.

Tests precondition:

- OVS (with DPDK) is started with OVS bridge/port added to configuration.

OVS/DPDK vhost tap configuration:

- OVS with DPDK is build/configured.

Collectd configuration:

```
<Plugin ovs_stats>
  Port "6640"
  Address "127.0.0.1"
  Socket "/var/run/openvswitch/db.sock"
  Bridges "br0"
```

```
</Plugin>
```

Table#4: OVS stats DPDK vhost interface tests.

#	Test summary	Steps	Expected	Test Result	Comments
1	Verify vhost OVS stats upon collectd restart.	<ol style="list-style-type: none"> 1. Start collectd. 2. Stop collectd. 3. Start collectd. 	<ol style="list-style-type: none"> 1. OVS stats updated, is same as in collectd plugin. 2. OVS stats not updated by collectd plugin. 3. OVS stats updated, is same as in collectd plugin. 		
2	Verify vhost OVS stats with different time interval.	<ol style="list-style-type: none"> 1. Change time interval to 1 second. Start collectd, testpmd. 2. Change time interval in range 2-90 seconds. Restart collectd. 	<ol style="list-style-type: none"> 1. OVS stats updated every second. 2. OVS stats updated every new defined time interval. OVS stats is same as in collectd plugin. 		
3	Verify vhost OVS stats with default configuration.	<ol style="list-style-type: none"> 1. Comment out "<Plugin ovs_stats>" section in collectd.conf. Start collectd, testpmd. 2. Define "collectd configuration" in collectd.conf file. Restart collectd service. 	<ol style="list-style-type: none"> 1. Collectd started with default configuration, check syslog. OVS stats updated every time interval by collectd. 2. Collectd started with given configuration. OVS stats are updated every time interval by collectd. 		

4	Verify vhost OVS stats for Rx packets/bytes counters.	<ol style="list-style-type: none"> 1. Start collectd. 2. Send traffic toward vhost port. 	<ol style="list-style-type: none"> 1. Collectd started without errors. 2. Vhost OVS statistic for Rx packets bytes is same as sent, shown by collectd plugin. 		
5	Verify vhost OVS stats for packets with different size.	<ol style="list-style-type: none"> 1. Start collectd. 2. Send packets toward vhost with different packet size [bytes]: 64, 128, 256, 512, 1024, 1518. 3. Try to send packets with size [bytes]: 60, 2048, 10000. 	<ol style="list-style-type: none"> 1. Collectd started. 2. Vhost OVS statistic for Rx packets sizes is same as sent, shown by collectd plugin. 3. Vhost OVS statistic for Rx packets sizes is same as sent, shown by collectd plugin. 		
6	Verify vhost OVS stats for unicast/multicast/broadcast packets.	<ol style="list-style-type: none"> 1. Start collectd. 2. Send packets toward vhost: unicast, multicast, broadcast. 	<ol style="list-style-type: none"> 1. Collectd started. 2. Vhost OVS statistic for unicast/multicast/broadcast is same as sent, shown by collectd plugin. 		
7	Verify vhost OVS stats upon OVS stop/start.	<ol style="list-style-type: none"> 1. Start collectd. 2. Send packets toward vhost. 3. Stop OVS. 4. Send packets toward vhost. 5. Start OVS. 6. Send packets toward vhost. 	<ol style="list-style-type: none"> 2. OVS stats updated, is same as in collectd plugin. 4. OVS stats not updated by collectd plugin. 6. OVS stats updated, is same as in collectd plugin. 		
8	Verify vhost OVS stats for new added OVS bridge, port.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create new bridge, port. <code>ovs-vsctl add-br new_bridge</code> <code>ovs-vsctl add-port br0 test_port1 -- set interface test_port1 type=internal</code> 3. Send packets toward vhost. 	<ol style="list-style-type: none"> 1. OVS stats updated for existing bridge/port, is same as in collectd plugin. 3. OVS stats updated for new bridge/port, is same as in collectd plugin. <p>ls /ovs_stats-br0.test_port1</p>		
9	Verify vhost OVS stats for removed port from bridge.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create bridge, port. 3. Send packets toward vhost. 4. Remove added port. 5. Send packets toward vhost. 	<ol style="list-style-type: none"> 3. OVS stats updated, is same as in collectd plugin. 5. OVS stats not updated by collectd plugin. 		
10	Verify vhost OVS stats for removed bridge.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create bridge, port. 3. Send packets toward vhost. 4. Remove added port, bridge. 5. Send packets toward vhost. 	<ol style="list-style-type: none"> 3. OVS stats updated, is same as in collectd plugin. 5. OVS stats not updated by collectd plugin. 		
11	Verify vhost OVS stats updated only for configured bridges.	<ol style="list-style-type: none"> 1. Start collectd. 2. Add one more bridge/port. 3. Send packets toward vhost. 4. Add just added bridge to collectd.conf "Bridges" section. 5. Send packets toward vhost. 	<ol style="list-style-type: none"> 3. OVS stats not updated by collectd plugin. 5. OVS stats updated, is same as in collectd plugin. 		
12	Verify vhost OVS stats updated over socket created by address/port pair.	<ol style="list-style-type: none"> 1. Stop openvswitch service. 2. Configure collectd.conf to work via address/port. Start collectd. 3. Emulate OVS server starting from zero values. 	<ol style="list-style-type: none"> 3. OVS stats updated, is same as in collectd plugin. 		
13	Verify vhost OVS stats updated upon overflow counters' values.	<ol style="list-style-type: none"> 1. Start python socket which emulates ovsdb replies. 2. Set OVS counter values: 0, 2^{63}, $2^{63} + 1$ (e.g. if <code>rx_errors</code> over). 3. Start collectd. 	<ol style="list-style-type: none"> 3. Values are shown in range 0 - $2^{63}-1$ for every counter. No errors or exceptions raised. 		

Collectd configuration:

```
<Plugin ovs_stats>
  Port "6640"
  Address "127.0.0.1"
  Socket "/var/run/openvswitch/db.sock"
  Bridges "br0"

</Plugin>
```

Table#5: OVS stats native (tap) vhost interface tests.

Note: See table #4 tests.

OVS events: native vhost interface (tap device) and DPDK vhost interface to retrieve link status.

Tests precondition:

- OVS (with DPDK) is started with OVS bridge/port added to configuration.

OVS/DPDK vhost tap configuration:

- OVS with DPDK is build/configured.

OVS events: DPDK vhost interface to retrieve statistic.

Collectd configuration:

```
<LoadPlugin python>
  Globals true
</LoadPlugin>
<Plugin python>
  ModulePath "path_to_plugin_file/"
  LogTraces true
  Interactive false
  Import "collectd-python_plugin"
</Plugin>

<Plugin ovs_events>
  Port 6640
  Address "127.0.0.1"
  Socket "/var/run/openvswitch/db.sock"
  Interfaces "br0"
  SendNotification true
  DispatchValues true
</Plugin>
```

Table#6: OVS events DPDK vhost interface tests.

#	Test summary	Steps	Expected	Test Result	Comments
1	Verify vhost OVS events upon collectd restart.	<ol style="list-style-type: none">1. Start collectd. Set link up /down/up of vhost port.2. Stop collectd. Set link up /down/up of vhost port.3. Start collectd. Set link up /down/up of vhost port.	<ol style="list-style-type: none">1. OVS link status is dispatched.2. OVS link status is not dispatched.3. OVS link status is dispatched.		

2	Verify vhost OVS events with default configuration.	<ol style="list-style-type: none"> 1. Comment out "<Plugin ovs_events>" section in collectd.conf. Start collectd. 2. Define "collectd configuration" in collectd.conf file. Restart collectd service. 3. Start collectd. Set link up/down/up of vhost port. 	<ol style="list-style-type: none"> 1. Collectd started with default configuration, check syslog. OVS link status is dispatched. 2. Collectd started with given configuration. 3. OVS link status is dispatched. 		
3	Verify vhost OVS events upon OVS stop/start.	<ol style="list-style-type: none"> 1. Start collectd. Set link up/down/up of vhost port. 2. Stop OVS. Set link up/down/up of vhost port. 3. Start OVS. Set link up/down/up of vhost port. 	<ol style="list-style-type: none"> 1. OVS link status is dispatched. 2. OVS link status is not dispatched. 3. OVS link status is dispatched. 		
4	Verify vhost OVS events with different time interval when SendNotification set to true.	<ol style="list-style-type: none"> 1. Set "SendNotification true" in collectd.conf "Plugin ovs_events>" section. 2. Change time interval to 1 second. Start collectd. 3. Set link up/down/up of vhost port. 4. Change time interval in range 2-90 seconds. Restart collectd. Set link up/down/up of vhost port. 	<ol style="list-style-type: none"> 3. OVS link status is updated every second. 4. OVS link status is updated every new defined time interval. 		
5	Verify vhost OVS events with different time interval when SendNotification set to false.	<ol style="list-style-type: none"> 1. Set "SendNotification false" in collectd.conf "Plugin ovs_events>" section. 2. Change time interval to 1 second. Start collectd. 3. Set link up/down/up of vhost port. 4. Change time interval in range 2-90 seconds. Restart collectd. Set link up/down/up of vhost port. 	<ol style="list-style-type: none"> 3. OVS events updated every second as statistic (tail -f /tmp/localhost/ovs_events-br0/gauge-link_status-YYYY-MM-DD) 4. OVS events updated every new defined time interval. 		
6	Verify vhost OVS events for new added OVS bridge, port.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create new bridge, port. ovs-vsctl add-br new_bridge ovs-vsctl add-port br0 test_port1 -- set interface test_port1 type=internal 3. Set link up/down/up for vhost added bridge/port. 	<ol style="list-style-type: none"> 3. OVS link status is dispatched for new bridge/port. 		
7	Verify vhost OVS events for removed port from bridge.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create bridge, port. 3. Set link up/down/up for added vhost port. 4. Remove added port. 5. Set link up/down/up for added vhost port. 	<ol style="list-style-type: none"> 3. OVS link status is dispatched. 5. OVS link status is not dispatched. 		
8	Verify vhost OVS events for removed bridge.	<ol style="list-style-type: none"> 1. Start collectd. 2. Create bridge, port. 3. Set link up/down/up for added vhost port. 4. Remove added port, bridge. 5. Set link up/down/up for added vhost port. 	<ol style="list-style-type: none"> 3. OVS link status is dispatched. 5. OVS link status is not dispatched. 		

9	Verify vhost OVS events are not sent when SendNotification set to false.	<ol style="list-style-type: none"> 1. Set "SendNotification false" in collectd.conf "Plugin ovs_events" section. 2. Start collectd. 3. Set link up/down/up for added vhost port. 	3. OVS link status is not dispatched. OVS stats/values are dispatched?		<p>MT: notification is not dispatched - but I think a stat/value is</p> <p>SA: Ok, to be verified.</p>
10	Verify vhost OVS events are sent when link to OVS database is lost.	<ol style="list-style-type: none"> 1. Set "NotifyLevel OKAY" for syslog plugin. 2. Set "SendNotification true". Start collectd. 3. Stop OVS: service openvswitch-switch stop. 4. Start OVS. 5. Stop OVS. 	2. 4. OVS link status notification message about loss connection to OVS DB recorded to syslog.		
11	Verify vhost OVS events are sent with defined be syslog plugin severity: OKAY, WARNING, ERROR.	<ol style="list-style-type: none"> 1. Set "SendNotification true". 2. Set "NotifyLevel OKAY" for syslog plugin. Start collectd. 3. Set link up/down/up for added vhost port. 4. Set "NotifyLevel WARNING" for syslog plugin. Restart collectd. 5. Set "NotifyLevel FAILURE" for syslog plugin. Restart collectd. 	<ol style="list-style-type: none"> 3. OVS link status notification message recorded to syslog. 4. OVS link status notification message recorded to syslog with Warning severity. 5. OVS link status notification message recorded to syslog with Failure (ERROR) severity. 		
12	Verify vhost OVS events metadata sent by collectd python plugin.	<ol style="list-style-type: none"> 1. Enable python (exec) plugin for handling metadata. 2. Start collectd. 3. Set link up/down/up for added vhost port. 	3. Metadata is present in notifications dispatched by collectd python plugin.		
13	Verify vhost OVS events monitored against interfaces set in collectd.conf.	<ol style="list-style-type: none"> 1. Set "Interfaces '<non-existent port>' ". Start collectd. 2. Set link up/down/up for added vhost port. 3. Set "Interfaces '<existent port>' ". Restart collectd. 4. Set link up/down/up for added vhost port. 	<ol style="list-style-type: none"> 2. OVS link status is not dispatched. 4. OVS link status is dispatched. 		
14	Verify vhost OVS events monitored against not defined interface section in collectd.conf.	<ol style="list-style-type: none"> 1. Comment out line "Interfaces '<port>' ". Start collectd. 2. Set link up/down/up for added vhost port. 	2. OVS link status is dispatched.		
15	Verify vhost OVS events upon DispatchValues set to false.	<ol style="list-style-type: none"> 1. Set "DispatchValues false". Start collectd. 2. Set link up/down/up for added vhost port. 3. Set "DispatchValues true". Start collectd. 4. Set link up/down/up for added vhost port. 	<ol style="list-style-type: none"> 2. OVS link status is not dispatched. 4. OVS link status is dispatched. 		
16	Verify vhost OVS events handle data about link status every 0.005 second.	<ol style="list-style-type: none"> 1. Set Interval to 0.005 seconds. Start collectd. 2. Set link up/down/up for added vhost port. 	2. OVS link status is updated every 0.005 seconds.		

OVS events: native vhost interface to retrieve statistic.

Collectd configuration:


```
<LoadPlugin python>
  Globals true
</LoadPlugin>
<Plugin python>
  ModulePath "path_to_plugin_file/"
  LogTraces true
  Interactive false
  Import "collectd-python_plugin"
</Plugin>

<Plugin ovs_events>
  Port 6640
  Address "127.0.0.1"
  Socket "/var/run/openvswitch/db.sock"
  Interfaces "br0"
  SendNotification true
  DispatchValues true
</Plugin>
```

Table#7: OVS events DPDK vhost interface tests.

Note: See table #6 tests.