

2022-03-16 ViNePerf Agenda and Minutes


Attendees

Sridhar Rao

Al Morton

Daniele Zulberti

Federica Paganelli

Sl. No	Topic	Presenters	Notes
1	Development Update		<p><i>Epic-VINEPERF-672: Create Tools as part of moselle release</i></p> <p>Epic-VINEPERF-671: Add support for newer software versions</p> <p>Epic-VINEPERF-670: Create clean workflows for Baremetal, Openstack and Kubernetes Usecases</p> <p>Epic-VINEPERF-669: Improve Stability for moselle Release</p> <p>Epic-VINEPERF-652: Enhance XTesting-ViNePerf Integration</p> <p>Task-VINEPERF-658: Enhance framework for XTesting-K8s Usecase</p> <p>Task-VINEPERF-654: XTesting-ViNePerf Integration Enhancement - Kubernetes</p> <p>Task-VINEPERF-653: XTesting-ViNePerf Enhancement - Openstack</p> <p>Epic-VINEPERF-638: Dataplane performance testing for various internal (within cloud) scenarios</p> <p>Task-VINEPERF-643: Pod-Pod Communication</p> <p>Daniele Zulberti to submit patch - reference deployment and config files (pod/nad), prox&trex.</p>
2.	Discussion of the results		<p>Summary :</p> <ol style="list-style-type: none">1. VPP performs better than OVS.2. Prox performs better than T-Rex3. Bi-Directional Prox is more inconsistent (throughput) than unidirectional4. Trex results are not consistent.5. Increasing hops, even with single node, affects the performance. Better summary for varying topologies6. Prox has some limitations - mainly w.r.t core assignments - which affects the performance. <ol style="list-style-type: none">1. VPP performs better than OVS<ol style="list-style-type: none">a. Bad OVS config?2. Prox performs better than T-Rex<ol style="list-style-type: none">a. Bad support for Trex execution in container?b. Bad resource allocation for Trex? few cores used?3. Bi-Directional Prox is more inconsistent (throughput) than unidirectional<ol style="list-style-type: none">a. Small queues for managing small packet sizes?4. Trex results are not consistent.<ol style="list-style-type: none">a. See point 25. Increasing hops, even with single node, affects the performance.<ol style="list-style-type: none">a. From PROX+VPP we can see that the latency is the most affected stat.b. For the other cases this could not be the major problem but just a side effect.6. Prox has some limitations<ol style="list-style-type: none">a. Can't increase core higher than 1 for each Prox's task - which affects the performance. (Is more research needed? Is it a configuration issue? Is it related to the container version of PROX?)7. Trex has some limitations<ol style="list-style-type: none">a. bidirectional traffic only (tried only with memif)b. driver net_virtio_user is not supported (No OVS testing)
3.	Limitation of L3		<div> Broken image</div> <p>Any of these two (red and green) paths even possible without manually adding routes/flows?</p> <p>Using IP-address is easy with iperf/netperf. Whereas, a Tgen (Prox/Trex), with DPDK-interfaces, it gets difficult.</p> <p>Not sure if our IPAM configurations are correct - Working on finding the right configuration (if any).</p> <p>How Userspace CNI handles IPAM configuration is not clear - with VPP, we didn't see any routes getting added.</p> <p>Reach out to Luc/Yuri - to check if they have achieved this with Prox. Maybe, Prox-L3swap mode can respond to ARPs.</p> <p>https://doc.dpdk.org/api/examples_2bond_2main_8c-example.html</p>