OVS Packet Processing

**Exact Match Cache**
- Logically, Single Table per datapath thread
- Exact Match
- 8192 entries / per thread

**Datapath Classifier**
- Logically, Single Table per datapath thread
- Wildcard Matches
- 65536 entries
- Each table is implemented as a priority list of subtables in order to implement wildcards

**Ofproto Classifier**
- Logically, Multiple (up to 255) Open Flow tables in pipeline per Open vSwitch bridge
- Wildcard Matches
- Each table is implemented as a priority list of subtables in order to implement wildcards

Cost of lookup increasing
OVS Flow Performance

#### Exact Match Cache
- Logically, Single Table per datapath thread
- Exact Match
- 8192 entries / per thread

#### Datapath Classifier
- Logically, Single Table per datapath thread
- Wildcard Matches
- Each table is implemented as a priority list of subtables in order to implement wildcards

#### Ofproto Classifier
- Logically, Multiple (up to 255) Open Flow tables in pipeline per OpenvSwitch bridge
- Each table is implemented as a priority list of subtables in order to implement wildcards

- rx cost
- lookup cost
- tx
- action
- cost
OVS Flow Performance Factors

Many factors affect the performance of each individual flow.
Topo-Phy
- X Mpps ingress
- Datapath can do Y million lookups per packet
- 1 lookup in switch per packet
- Y Mpps egress

VMs in Parallel
- X Mpps ingress
- X/2 Mpps into each VM
- Datapath can do Y million lookups per packet
- 2 lookups in switch per packet
- ~Y/2 Mpps egress
- ~Y/4 Mpps per VM

VMs in Series
- X Mpps ingress
- Datapath can do Y million lookups per packet
- 3 lookups in switch per packet
- ~Y/3 Mpps egress

Topology of a use case can affect performance as perceived externally from compute node
Openstack Neutron Configuration Example

Can overlap IP and MAC addresses across tenants

Not showing security groups configuration

```
[tester@slpixx00381991 openstack_test]$ sudo ovs-ofctl dump-flows br-int
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=1522.667s, table=0, n_packets=1524, n_bytes=135142, idle_age=985, priority=1
actions=NORMAL
cookie=0x0, duration=1322.207s, table=0, n_packets=0, n_bytes=0, idle_age=1322,
priority=3, in_port=1, dl_vlan=1002 actions=mod_vlan_vid:3,NORMAL
cookie=0x0, duration=1353.799s, table=0, n_packets=0, n_bytes=0, idle_age=1355,
priority=3, in_port=1, dl_vlan=1000 actions=mod_vlan_vid:2,NORMAL
cookie=0x0, duration=1522.596s, table=0, n_packets=0, n_bytes=0, idle_age=1522, priority=2, in_port=1
actions=drop
cookie=0x0, duration=1522.663s, table=23, n_packets=0, n_bytes=0, idle_age=1522, priority=0
actions=drop
```

```
[tester@slpixx00381991 openstack_test]$ sudo ovs-ofctl dump-flows br-tun
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=1683.106s, table=0, n_packets=0, n_bytes=0, idle_age=1683, priority=1
actions=NORMAL
cookie=0x0, duration=1482.605s, table=0, n_packets=186, n_bytes=19092, idle_age=1241,
priority=4, in_port=2, dl_vlan=3 actions=mod_vlan_vid:1002,NORMAL
cookie=0x0, duration=1516.269s, table=0, n_packets=180, n_bytes=18732, idle_age=1313,
priority=4, in_port=2, dl_vlan=2 actions=mod_vlan_vid:1000,NORMAL
cookie=0x0, duration=1683.055s, table=0, n_packets=144, n_bytes=12012, idle_age=1311,
priority=2, in_port=2 actions=drop
```

```
[tester@slpixx00381991 openstack_test]$ sudo ovs-ofctl dump-flows br-ex
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=1593.857s, table=0, n_packets=66, n_bytes=7344, idle_age=1265, priority=0
actions=NORMAL
```
How can we document performance requirements?

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Topology</th>
<th>Action Types</th>
<th>Number of Flows</th>
<th>Flow table occupancy rate</th>
<th>.. etc ..</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vEPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.. etc ..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• “Baremetal-Like Performance” is not sufficiently specific to target optimization efforts. Therefore more detail is necessary

• Above shows one example, however:
  • Too complicated?
  • Exposes operator’s confidential information?
How can we document performance requirements?

<table>
<thead>
<tr>
<th>Use Case</th>
<th>vsperf test case</th>
<th>LTD.x</th>
<th>LTD.y</th>
<th>LTD.z</th>
<th>.. etc ..</th>
<th>.. etc ..</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vEPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.. etc ..</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Specific requirement
- Easily testable
- Still exposes some confidential information?
How can we document performance requirements?

<table>
<thead>
<tr>
<th>Use Case</th>
<th>vsperf test case</th>
<th>LTD.x</th>
<th>LTD.y</th>
<th>LTD.z</th>
<th>.. etc ..</th>
<th>.. etc ..</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 0% use cases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 33% use cases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; 66% use cases</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Landing zone requirements
- Specific requirement
- Easily testable