Prague, Nov 19th, 2025

Uncovering Testing Gaps in OVN Incremental Processing

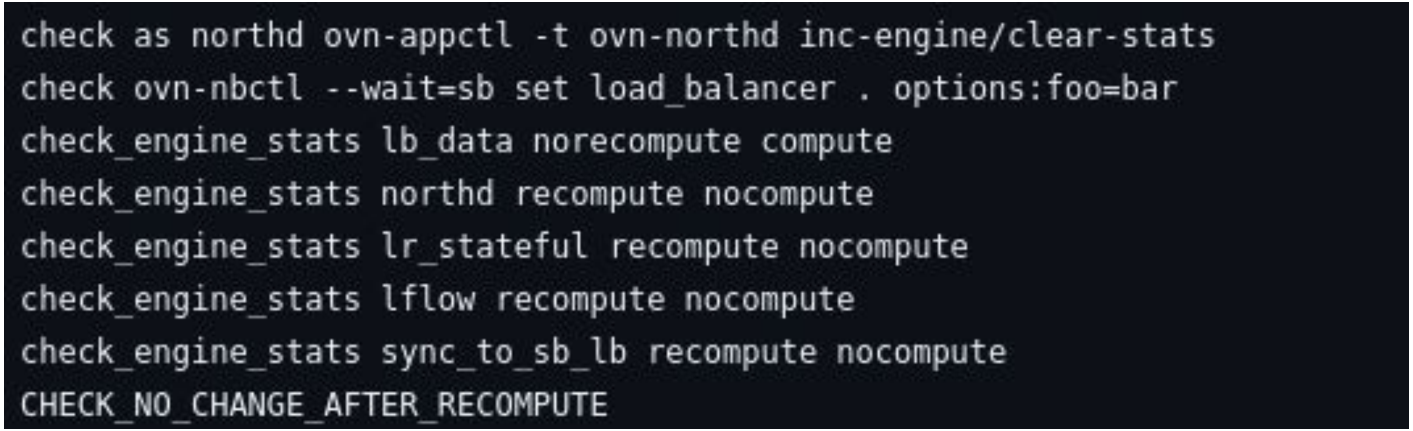
Jacob Tanenbaum, Red Hat

Background

- Updates to OVN are processed in two ways
  - recalculation of the whole database
  - The incremental processing engine recalculating only related objects.
- Database recalculation was the first way OVN processed updates so it is the standard which incremental processing is tested against

The Setup

- Testing of the incremental processor is done very simply



```
check as northd ovn-appctl -t ovn-northd inc-engine/clear-stats
check ovn-nbctl --wait=sb set load_balancer . options:foo=bar
check_engine_stats lb_data norecompute compute
check_engine_stats northd recompute nocompute
check_engine_stats lr_stateful recompute nocompute
check_engine_stats lflow recompute nocompute
check_engine_stats sync_to_sb_lb recompute nocompute
CHECK_NO_CHANGE_AFTER_RECOMPUTE
```

The Issue

- We are only testing the incremental processor’s ability to process updates against a freshly recalculated database.
- The incremental processor relies on the state of internal ovn structs and the databases but we don’t test for the correctness of them.
- As more and more updates are processed using the incremental processing engine it becomes more and more important that the state of internal objects are left in the correct state after processing processing the update
- These issues will be very difficult to spot

Improvements

- Store the output we are expected to produce for each step of each test
- Run each test twice once to force recalculate and gather the expected results and the second time to compare against the incremental processor results
- Implement unit testing to check the validity of the internal data
- A few very long test cases that do not force recompute but do check the recompute counters while performing a larger number of transactions.

Slide Title

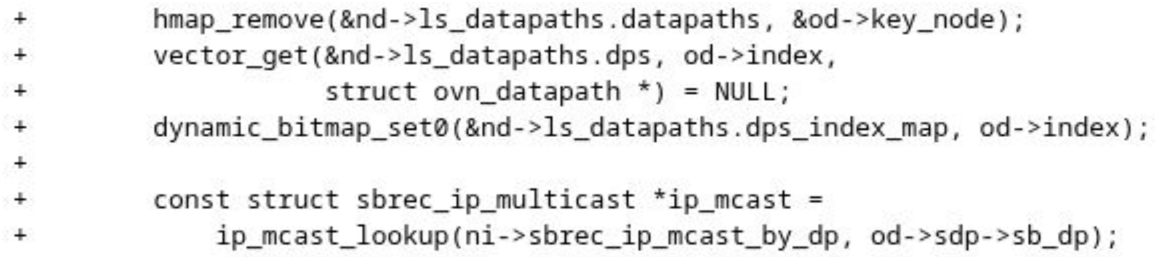
Any Questions?

Thank you

For questions/concerns/corrections please contact me through email jtanenba@redhat.com

The Issue (cont)

- These issues are going to be hard to spot



```
+       hmap_remove(&nd->ls_datapaths.datapaths, &od->key_node);
+       vector_get(&nd->ls_datapaths.dps, od->index,
+                  struct ovn_datapath *) = NULL;
+       dynamic_bitmap_set0(&nd->ls_datapaths.dps_index_map, od->index);
+
+       const struct sbrec_ip_multicast *ip_mcast =
+           ip_mcast_lookup(ni->sbrec_ip_mcast_by_dp, od->sdp->sb_dp);
```