Trigger Based VNF Migration in a Data Centre

Kurien Zacharia
143059002
IIT Bombay

Bhavesh Singh
143059003
IIT Bombay

Vikas Kumar
163059007
IIT Bombay

I. Problem Statement

- Model a given IP based network in a data-center with appropriate VNFs.
- Use YANG to model trigger based migration of VNFs to a different host.

II. Tools Used

We have used the following tools in our project

1) Mininet Network Simulator with *ovsk* switch
2) OpenDayLight Controller (Boron SR3)

III. Solution Steps

We have divided the problem into folllowing subsections as given below.

A. Creating a Datacenter Topology

We have defined a *k*-ary Fat Tree Data Centre Toplogy as showin in figure 1 using Python API and fetched it to mininet as custom toplogy with *k*=4 having 20 switches, 16 hosts and 48 links. We have used Fat Tree because it has identical bisection bandwidth.

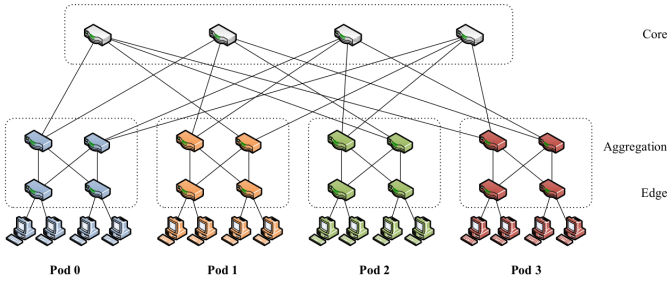

Fig. 1. Fat Tree Topology Source:[1]

B. Mininet

We have instantiated the custom Data Centre Toplogy in mininet using the `--custom` option and added a remote Open Day Light Controller which was running on the remote host using the `--controller` option into mininet. Then we have manually associated a queue with the port using ovs-vsctl set port qos command. We are running `SimpleHTTPServer` inside a mininet host and using the `httperf` - HTTP performance measurement tool to generate load on the server. OVS does not support OpenFlow 1.3 meters. So for the rate limiting QoS feature, we have implemented a queue on the port to which the Server is connected.

C. Install Flows

To install the flows to the OpenFlow switches we have used the northbound REST API of controller.

D. Opendaylight MD-SAL Application

To create an MD-SAL application, the following steps are used.

1) Define a YANG topology.
2) Compile the YANG model to generate a RESTCONF API.
3) Add the code to the generated Java bindings to register listeners for the model changes.
4) The code will install flows with queues to the switch for monitoring.
5) Attach a listener to the openflow queue-statistics and calculate the rate. If the rate exceeds a certain threshold, trigger the migration of the VNF.

Our VNF was just a simple http server on a host in mininet. The migration was emulated by stopping the server on a host and starting it on another host.

IV. Problems Faced

We have faced the following issues while implementing the above solution steps.

1) The mininet toplogy was not converging in ODL Controller. Later we have figured out that it was due to naming convention associated with the switch names in mininet which took most of our time.
2) Openflow Version Negotiation failed problem between ODL and Mininet. The default version supported with ODL Boron release was OF1.0 so we have changed to OF1.3 using the custom.properties file inside ODL. While running the ovsk switch with OpenFlow1.3 protocol, the version negotiation problem arised as we have not specified the protocols option with the ovs-ofctl command to retieve the flows.
3) NO meter table support in *ovsk* switch. We wanted to implement OpenFlow Meter Table as per OpenFlow 1.3 Specification for rate limiting the flows but later we have figured out that there was some implementation bug in *ovsk* switch as max meters was set to 0 as retrieved from operational datastore of controller. So

when we tried to install the meter into switch using REST API it was not pushed into operational datastore and remained in the config datastore only. Later we tried to implement per port queues for QoS in the OpenFlow Switches.

V. Conclusion

1) YANG model defining the parameters for VNF migration was created
2) Compiled YANG model and generated the REST API's
3) Implemented MDSAL listener to listen for changes to the DataStore
4) Partially able to push flows to the swtich from the MDSAL layer (without queue options)
5) Manually create and push flows along with queue using REST conf
6) Simulated HTTP Server as VNF along with a watcher process which can migrate the server across mininet nodes
7) Use httperf as a load generator to simulate traffic flow

References

[1] http://sdn-in-datacenters.blogspot.in/2014/04/literature-survey-portland-design.html