OPENSTACK NETWORKING PLUGINS: OPTIMIZING PERFORMANCE
PROJECT PROPOSAL

1. Vision of the Project:

Software based networking infrastructure is a major component of cloud infrastructure along with cloud compute and storage infrastructure. Workloads such as cryptography, deep packet inspection (DPI) and delay sensitive traffic transmission (such as video and financial data) require high performance networking. OpenStack provides cloud based networking services (layer 3 networking services includes NAT, static routing, etc.). Some OpenStack use cases require highly performant networking services. Therefore, Cisco has provided OpenStack Plugin support to offload networking services to specialized networking hardware. However, there is little independent data analyzing and comparing performance of the reference OpenStack networking implementation versus these networking plugins. The vision of the project incorporates testing and comparisons of the reference OpenStack solution with the Cisco plugin solution.

2. Goals of the Project:

- The primary goals incorporate familiarization with Devstack and OpenStack, Horizon & CLI , Create router-create network-attach interface to router.
- Gain understanding of Ethernet, VLAN, routing, HTTP protocol, REST, tcpdump, Linux network namespace, openvswitch, Plugin, Agent, Driver concepts,etc
- Python: Work on Pycharm for major parts of the project
- Familiarize with Performance tools such as iperf, vmtp and Kloudbuster
- Bring up an ASR1K and execute Performance Test Plan on the community

Users/Personas Of the Project:

Optimizing the performance of these networking plugins targets end-users by increasing the reliability and availability of OpenStack cloud environments.

The proposed scheduling model will be utilized by Cisco in order to adjust or make any performance related enhancements.

3. Scope and Features of the Project:

The features and scopes of the project include:-

- Setting up of Virtual Machine, Devstack environment for OpenStack.
- Open Neutron as project, connect to Neutron database and inspect tables and their content. Running neutron server in Pycharm.
- Gain basic understanding of Ethernet,VLAN, routing, HTTP protocol, REST,tcpdump, Linux network namespaces,openvswitch

- Familiarization of Plugins,Agent, Driver concepts, Security Groups
- Analyzation of traffic from Inside to Outside and Outside to Inside
- Understand the packet path from a Virtual Machine to another Virtual Machine on the same subnet
- Get a basic understanding of Performance tools such as iperf , vmtp and Kloudbuster.
- Understand the ASR1K Plugin and Configure Agent differences from the Community Implementation
- Follow the code for a simple workflow and execute a test plan on the community
- Get an overview of network traffic testing

4. Solution Concept

The current architecture of Neutron could limit scalability, reliability and performance of the cloud. The project entails gaining an understanding of performance tools in order to determine how the Cisco ASR 1000 can overcome these limitations. Our results will lead to possible recommendations for Cisco to make any adjustments in design and utilization.

Global Architectural Structure of the Project and a walkthrough:
The OpenStack system, the limitations of Neutron and the solution through Cisco's networking plugin and the project goals will be outlined below.

OpenStack is an open source cloud operating system that controls compute, storage, and networking resources throughout a datacenter. Administrators and users manage it through a dashboard to provision services through a web interface. The core services are Keystone(authentication service), Nova (compute service), Glance (image service), Cinder and Swift (storage) and Neutron (networking service).

Currently, there are limitations of Openstack Neutron, which establishes open networking services for the cloud. In Neutron, packets are routed through the Neutron hosts demonstrated below in Figure[1]. As the OpenStack cloud gets larger, traffic through the hosts increases causing the network to become a bottleneck. These bottlenecks can occur in several ways, ie. in establishing current networks. Capacity barriers limit cloud expansion, hence reducing user satisfaction with OpenStack.

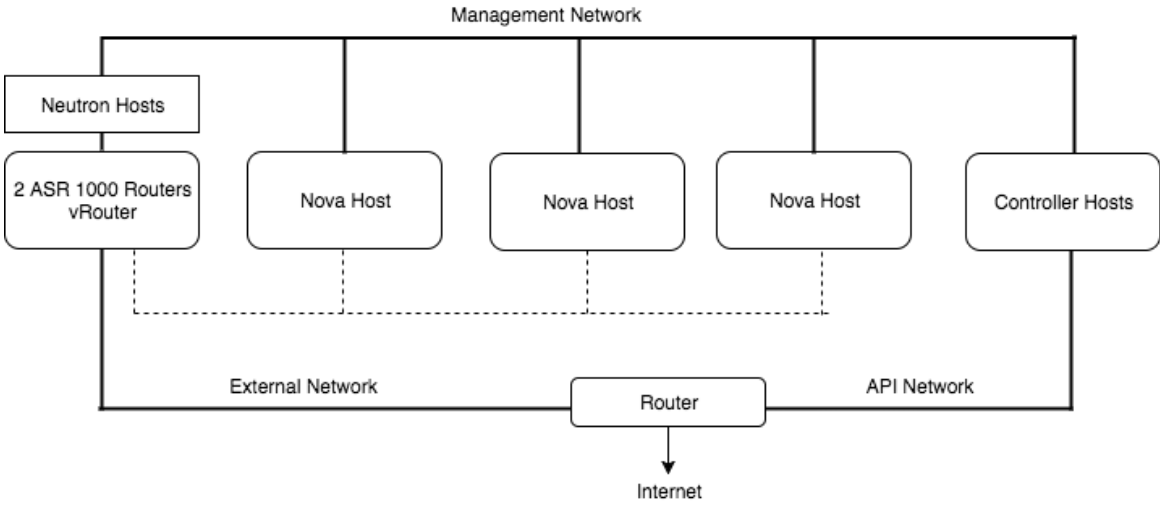

Figure [1]: Diagram of network architecture in OpenStack. Task are filtered to the ASR 1000 routers in an effort to reduce network traffic and increase external network availability.

Cisco solved the issue using a network plugin that moves tasks from network hosts to a pair of ASR 1000 routers. Cisco® ASR 1000 Series Aggregation Services Routers aggregate multiple WAN connections and network services including encryption and traffic management, and forward them across WAN connections at line speeds from 2.5 to 200 Gbps. With Cisco's solution, it is possible to increase external network connectivity, tenant routing, tenant networks, data throughput, datapad redundancy in OpenStack cloud.

There will be load testing using both the CISCO plugin and the default Neutron plugin for various applications with varying sizes of data. Depending on the outcome of the tests we will create a set of recommendations for the determining which jobs are suitable to be routed through the ASR1k hardware via the CISCO plugin and which jobs can be run through Neutron without specialized hardware support.

5. Minimum Acceptance Criteria

Minimum acceptance criteria is a functional OpenStack environment using devstack, using PyCharm to develop tests on ASR1k and compare its performance for various applications with Neutron, the default Openstack Virtual Routing service.
Stretch goals are:

- Create a scheduling model of hardware vs. software
- Utilize other performance tools such as iperf, vmtp, and Kloudbuster
- Analyze the Community implementation

6. Release Planning

Release #1 (First Sprint):
The first sprint will focus on the basis of setting up a basic OpenStack environment.

- Install Devstack on a virtual machine of choice (VMware Fusion, Virtualbox).
- Test out Horizon, OpenStack's UI, and command line interface (CLI)
- Exploring common workflows, from environment specific workflows to release specific workflows
- Create a router and a network
- Install PyCharm
- Open neutron as a project
- Connect to neutron DB and inspect tables and their content

Release #2 (Second Sprint):
The second sprint shall focus on trying out common workflows and get familiar with some performance tools.

- Familiarize with Ethernet, VLAN, routing, HTTP protocol, REST, tcpdump, Linux network namespaces,openvswitch
- Get an understanding on Plugins,Agent and Driver concepts
- Create VM, Security Groups and attach floating IP
- Understand basic idea behind NAT
- Adding breakpoints in the neutron server and Agent and follow the code for one of the workflows
- Use Pycharm for running neutron server, also perform remote debugging using Pycharm
- Analyze performance : Latency versus Throughput
- Familiarize with performance tools such as iperf, vmtp and Kloudbuster

Release #3 (Third Sprint):
This sprint deals with the Cisco Plugin ASR1k

- Work on the devstack environment to bring up an ASR1k
- Logging in to the ASR and looking at basic configuration
- Use IOS XE CLI
- Repeat the common workflows and check impact on ASR configuration
- Gain an understanding of the ASR1 Plugin and Config Agent
- Analyze the differences from the Community Implementation

Release #4 (Fourth Sprint):

- Start the execution of the test plan on the community. The implementation is followed by the ASR1K environment

Release #5 (Fifth Sprint):

- Continuation of the Performance test plans on the community

Scheduling Model recommendations/Code enhancements:-

- Scheduling model: Some tools could be additionally developed to influence the scheduling of hardware vs software based implementations based on the workload inputs.
- Work or help with any performance related enhancements with the mentor team members.