An Analysis of SDN-OpenStack Integration

Olena Tkachova
Telecommunication system Department
Kharkov National University of Radio Electronics
Kharkov, Ukraine
korov4enko@mail.ru

Mohammed Jamal Salim,
Abdulghafoor Raed Yahya
Telecommunication system Department
Odessa National Academy of Telecommunications
named after O.S. Popov
Odessa, Ukraine

***Abstract*— This paper is devoted to analysis of integration of SDN solutions to the OpenStack. Architecture of networking OpenStack module or Neutron and plug-ins for communication with SDN is considered. Different types of SDN controllers also considered in the paper. An experimental model of SDN-OpenStack integration was created according to the solution. Such characteristics of SDN-OpenStack solution as delay, max amount of UDP and TCP flows obtained in the experiment.**

Keywords—OpenStack; Neutron; SDN controller; performance; cloud technology

I. Introduction

Today OpenStack technology gained the great popularity among cloud-computing management platforms. However, convergence of OpenStack technology and traditional networks does not always bring the desired results. An inefficient load balancing and traffic filtering, data transmission delays, a performance of different elements are the main causes of problems.

The paradigm of Software-Defined networking (SDN) often is used for organizing a correct communication between elements of physical and virtualized nature. The main feature of SDN is centralized management and monitoring, which are carried out by a single network element – controller [1]. The advantage SDN integration into cloud technologies are advanced management, seamless convergence and optimal load balancing.

Lack of documentation, formal recommendations and requirements for methods of interaction of cloud technologies and SDN elements lead to the appearance a number of errors. Today, many companies provide in their solutions SDN-OpenStack integration into existing technologies: Cisco, HP, Juniper, NEC, IBM, etc [2]. Network operating system (NOS) is the basis of SDN controllers. Different NOSs have different network characteristics and communication algorithms.

Modeling and analyzing of interaction between SDN and OpenStack elements makes it possible to determine the functionality and performance for different types of controllers. The result of analysis will allow to choose the optimal solution which can be applicable to support future collaborative of SDN-OpenStack solutions.

II. Neutron Architecture

All network services and applications have a wide range of network requirements. At the same moment, many distribution aspects depend on network equipment. Providers can use network equipment (NE) and technology from different vendors. OpenStack uses Neutron module to complicate cloud platform with another technology [3]. Functional element OpenStack Networking or Neutron collect a variety of plug-ins that provide successful integration with other NEs. Neutron provides the tenants of cloud resources, API-interface through which they can configure and flexible policies to create complex network topologies, for example, organize support of multilevel Web-servicers.

Neutron based on abstraction model that include virtual networks, subnetworks, ports. Network is an isolated segment of the link layer (L2) that is very similar to VLAN. Neutron main components are [3]:

Neutron-server— it is the primary server of OpenStack Networking (Python daemon) redirects the API-interface request to configured OpenStack Networking plug-ins. Cloud environments administrator chose the plug-in that is the most suitable for this technology.

In addition, the structure of OpenStack Networking includes three agents that interact with the main Neutron process through message queue or through a standard API-interface:

- Neutron-dhcp-agent provides a DHCP-services (Dynamic Host Configuration Protocol) to all endpoints.
- Neutron-l2-agent support L2 functionality for virtual machines to an external network.
- Neutron-l3-agent functionality supports L3/NAT, to allow VMs access to the external network.
- Optional agent determined by plug-in (neutron-*-agent) configures a local virtual switch on each hypervisor.

Figure 1 shows the architecture of OpenStack Neutron component and their interaction with OpenStack Compute (Nova). Bold lines highlight the process of interaction.

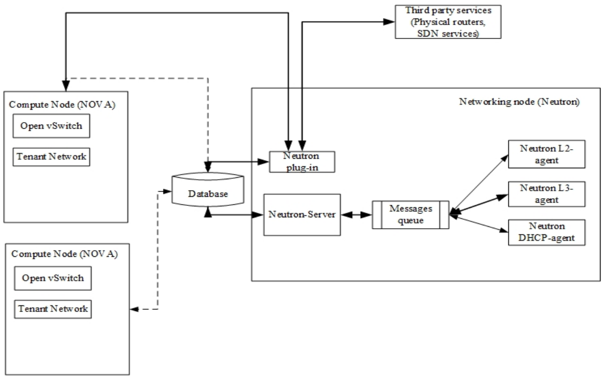

Fig. 1. Neutron architecture

Neutron module constantly is updated. However, it has several significant disadvantages and limitations compared to physical network equipment. The main disadvantages of OpenStack Networking are not flexible cooperation in the transport layer, impossible to support for some types of routers, limited scalability, VLAN configuration limitations, limited use firewall functions and NAT [4].

Applying the Software Defined Networking concept and its integration with OpenStack technology can eliminate a number of disadvantages. In this case routing and traffic isolation function that perform SDN controller allow to speed up the data transfer process (as compared with conventional methods of traffic control), and eliminate the Neutron necessary to know information about physical topology of the network. SDN controller integration in OpenStack also allows to increase scalability through effective load balancing.

III. SDN – OpenStack interaction model

SDN is a network concept that allows centralized control plane to manage the entire data plane, so that the network operators and providers can control and manage their own virtualized resources and networks [5]. SDN also provides API communication between the hardware and the operating system, and between network elements and operating system [6].

The SDN controller is responsible for providing complete information about network topology and its status, monitoring and management, transmission of information about changes to the end-users (VMs).

The change to the underlying network comes from network applications (Neutron API) and running to the controller operation systems using the Northbound API [6]. The controller processes the new information and provides further management data plane layer. The SDN controller manages the underlying hardware (hardware devices and VMs) via Southbound APIs using OpenFlow and OF-config protocols [5]. The Figure 3 below gives a general idea of SDN controller integration into OpenStack.

Integration of SDN controller and OpenStack Networking Neutron by using plug-ins that provide centralized network management. Many plug-ins with different functions and operating parameters exists on both side: SDN and Neutron. The following Neutron plug-ins are currently available: Open vSwitch, Cisco UCS/Nexus, Linux Bridge, Modular Layer 2, REST Proxy Plugin, RUY. SDN controllers such as OpenDaylight, Ryu, and Floodlight, NOX use this plug-ins to organize interaction between Neutron and SDN [2, 7].

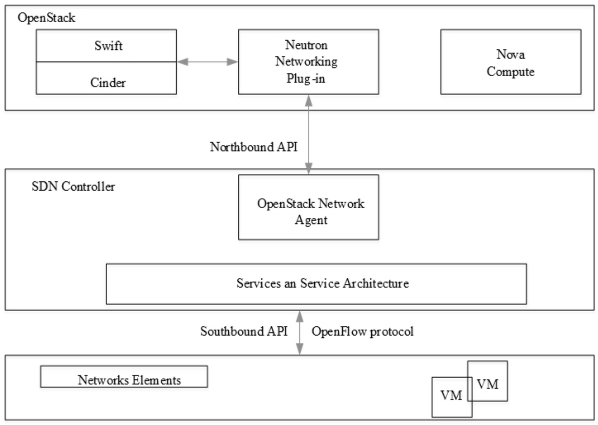

Fig. 2. Integration of SDN controller to the OpenStack

For example, OpenDaylight controller communicates with OpenStack Networking by using Modular Layer-2 plugin present. RYU controller uses special RYU plugin present on the Netron module. Flood Light NOS communicate with OpenStack Networking by using REST Proxy Plugin.

IV. Analyzing Results

Unfortunately, today there are not enough sources with documentation about integration SDN into OpenStack. The lack of formal description and vendor's test reports don't allow to choose the best practice solution for SDN controller integration into OpenStack. As written above, SND concepts separates control plan (management algorithms) from data plan (data transferring). In this way should be formalized and analyzed control plan and data plan characteristics.

The model of interaction that shown on Figure 3 is considered as a fragment of experimental network. The model allows assessing the effectiveness and performance of SDN-OpenStack solutions.

Throughput, scalability, delay were considered to analyze a data plan characteristics with SDN integration and without SDN. Network that uses OpenStack nova-network for networking and Linux bridge module for communication between VMs on a host server was considered as a baseline solution. Two VMs were used to evaluate delay, bandwidth and fault tolerant.

For communication between two virtual machines using Linux-bridge (baseline scenario) and different SDN controllers, for communication between network fragments used L2 tunneling.

Multi node OpenStack infrastructure has a compute node on one server, and controller, neutron, nova components, glance, on another server. We allocated resources (RAM, Storage, CPU) based on the functionality of the nodes.

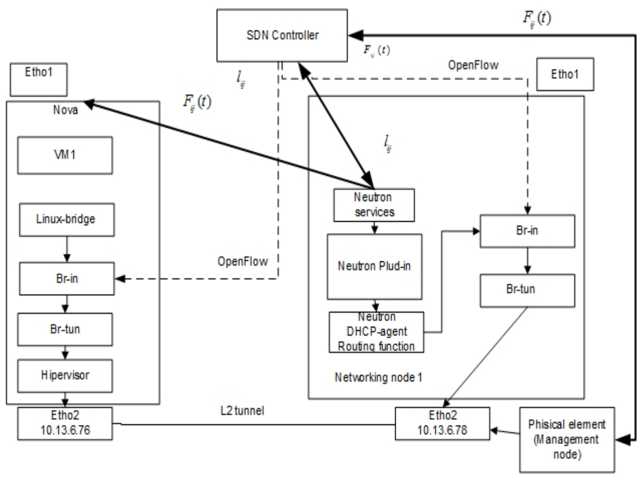

Fig. 3. Experimental model

Table 1 shows the results of different SDN-OpenStack integration solutions.

TABLE I.

	Open Daylight	*RUY*	*Flood light*	*Linux bridge*
Average latency	10,48 ms	2,9 ms	5,4 ms	1,3 ms
The average transmission delay	4,3 ms	2,7 ms	1.56 ms	0,403 ms
The maximum number of TCP packets for various MSS (1490 bytes, 9000 bytes)	MSS 1490: 18Mbps MSS - 9000: 17 Mbps	MSS 1490: 291 Mbps MSS - 9000: 293 Mbps	MSS 1490: 112 Mbps MSS – 9000: 487 Mbps	MSS 1490: 107 Mbps MSS – 9000: 454 Mbps
The maximum number of UDP packets with different number of parallel section	P = 1 session 1,05Mbps, 0% loss P = 3 sessions 480 Kbps, 2% loss	P = 1 session 21,9Mbps, 30 % loss P = 3 sessions 2,1Mbps, 10 % loss	P = 1 session 28,4 Mbps 27% loss P = 3 sessions 3,31Mbps 15% loss	P = 1 session 996 Mbps 46% loss P = 3 sessions 2,01 Mbps 44% loss
Maximum Allowed TCP flows	100 flows/sec - 0 % loss 1000 flows/sec - 3 % loss 10000 flows/sec - 98 % loss	100 flows/sec - 0% loss 1000 flows/sec - 7% flows/sec 10000 flows/sec - 100 % loss	100 flows/sec - 0% loss 1000 flows/sec - 10% flows/sec 10000 flows/sec - 100% loss	100 flows/sec - 0% loss 1000 flows/sec - 5% flows/sec 10000 flows/sec - 8% loss

To obtain the main network characteristics the following commands were used [8]:

Average latency for data transmission between two network nodes (ms)
$ for i in seq 1 20; do
ping -c 1 -q 10.10.should be 10.2

The average transmission delay (ms)
$ ping -c 50000 -f 10.13.6.76

The maximum number of TCP packets for various MSS (1490 bytes, 9000 bytes)
$ for i in 1490 9000; do
iperf -c 10.13.6.76 -t

The maximum number of UDP packets with different number of concurrent sessions (1 and 3)
$ for i in 1 3; do
iperf -c 10.13.6.76 -u -t 60 -b 10G –P

Max allowed TCP flows between known hosts, without zero drops
$ For i in 10 100 1000 10000; do
hping3 -c `expr $ i \ * 30` -iu`expr 1,000,000 / $ i` -S -q -I -p ++ 1024 10.13.6.76

Analyzing the results of the experiment revealed that the integration of SDN in OpenStack architecture use different controllers leads to different results. The data presented in the table shows that OpenDaylight has inferior performance to the delay time and the maximum data rate than Flood light, RYU and baseline solution. Nevertheless, OpenDaylight showed a high level of resiliency, and therefore the reliability is higher than RYU and Flood light. POX does not support integration with OpenStack at all.

References

[1] Architecture SDN [Electronic resource] // Open Networking Foundation. — [2014]. — Mode of access: https://www.opennetworking.org/ 2.

[2] Software-Defined Networking: The New Norm for Networks [Electronic resource] // Open Networking Foundation. — [2015]. — Mode of access: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[3] OpenStack Cloud Administrator Guide [Electronic resource] — [2015]. — Mode of access: http://docs.openstack.org/admin-guide-cloud/index.html

[4] A 10 Networks Openstack Integration [Electronic recource] // Customer Driven Innovation — [2015]. — Mode of access: https://www.a10networks.com/sites/default/files/resource-files/A10-SB-19107-EN.pdf

[5] Devlic, A., Ericsson, R., John. A Use-Case Based Analysis of Network Management Functions in the ONF SDN Model // European Workshop on Software Defined Networking,Vo: 10.1109/EWSDN.2012.11 – 2012 – pp. 85 – 90

[6] Myung-Ki, S., Ki-Hyuk, N., Hyoung-Jun, K. (2012). Software-defined networking (SDN): A reference architecture and open APIs. International Conference on ICT Convergence (ICTC), Vo: 10.1109/ICTC.2012.6386859 – 2012 – pp. 360-361

[7] OpenDaylight Integration [Electronic resource] — [2015]. — Mode of access:http://openstack.redhat.com/OpenDaylight_intergration

[8] Test suite for SDN-based Network Virtualization [Electronic resource] // SDN Hub — [2015]. — Mode of access: http://sdnhub.org/projects/sdn-test-suite/