Integrated Intelligent Research (IIR)

International Journal of Computing Algorithm
Volume: 06 Issue: 01 June 2017 Page No.51-54
ISSN: 2278-2397

Software Framework for Enterprise Solutions

Faizal Hajamohideen, Karthikeyan.S and Noushath.S
Department of Information Technology College of Applied Sciences Sohar – Sultanate of Oman
Email: faizalh@cas.edu.om, karthikeyan@cas.edu.om, noushath.soh@cas.edu.om

Abstract - A free and open source programming Language which is not only reliable and stable but can also implement wide range of application domains is of paramount importance to an enterprise. Python is such a general-purpose programming language that is widely used among various researchers/academicians/ industries. Availability of rich resource of library makes this programming language a widely preferred one. Python's open source libraries provide comprehensive support for development of applications such as web development, data analysis, artificial intelligence, scientific computing, desktop apps and utility tools. In this paper, we discuss general features of Python programming languages along with its support for packet manipulation and open v Switch configuration.

I. INTRODUCTION

Python is free and open-source software which is widely used by software developers and researchers. This programming tool is managed by the non-profit organization known as Python Software Foundation. The salient features of Python programming language are that it's a high-level, interpreted, dynamic and general-purpose programming language. One of the main advantages of Python programming is that it supports multiple programming paradigms such as object-oriented, imperative and procedural programming styles. The syntax of Python programming emphasizes on code readability. Unlike other programming languages such as C++ or Java, Python's syntax allows the user to represent a concept in fewer lines of code.Python interpreters are available for all platforms. Python supports cross-platform paradigm meaning that an application developed in one platform can be used as it is in other platform without the need of the Python interpreter. Python supports command-line interface or shell through which user can enter statements sequentially and results can be obtained immediately. In addition, it also provides ways to write a complete script and run it using its interpreter.

One of the main strengths of Python programming is it has a huge standard library supporting wide variety of functionality such as graphical user interface, web framework, connecting relational database, networking and communications. It is also a popular tool in test frameworks, automation and system administrations. For more than a decade now, Python has been a go-to choice for variety of processing, machine learning etc.On the other hand, there is a proprietary software equivalent to Python programming which is still a popular choice for software developers and researchers world-wide. However, Python should be preferable to its proprietary counterpart for various prominent reasons such as:

1) One can write more compact and readable code in Python than a Matlab version.
2) As mentioned earlier, Python is free and open whereas Matlab programming is hugely expensive.
3) Compared to Matlab, Python adheres to de-facto stan-dards of programming such as zero indexing, use of square brackets for indexing etc. This is very convenient for one who is beginning to use Python or converts code from other platform to Python and vice-versa, work across variety of programming platforms.
4) Unlike many standard programming languages such as C, C++, Java, Matlab; Python supports variety of data structures such as List, Tuple, Dictionary etc. which are considered to be highly efficient in various implementations.

For more technical information and for the download of the interpreter, visit the official webpage [1]. In this paper, we discuss general features of python programming language and its competitiveness to the contemporary proprietary based software systems. Furthermore, we present a case study that describes implementation of packet manipulation tool using Python codes. In addition, the Python API compatible Open vSwitch to replace a switching stack for hardware virtualization environments as a advanced multiple protocols support and standards used in computer networks are presented. The rest of the paper is organized as follows: Section II describes a tool known as Scapy written in Python language for interactive packet manipulation. Section III discusses the implementation features of Open vSwitch using Python package distribution. Section IV briefly describes other free and open source software available similar to that of Python. Finally, conclusions are drawn in section V.

II. INTERACTIVE PACKET MANIPULATION TOOL

Scapy is a packet manipulation tool, developed by Philippe Biondi, Guillaume Valadon and Pierre Lalet in August 2007. This tool is written in Python language, which is an open source. The latest version 2.3.2 was released on January 2016 [2].It has used to forge or decode the packets of various other network protocols such as TCP, UDP, IP and so on. Scapy is handling various security related functions like scanning, trace routing and network discovery. The uniqueness of this tool is ending an invalid frame, injecting customized 802.11 frames, VOIP decoding in the customized network. The Scapy provides general functionalities to the network applications rather than specific functions provided by various other tools like Nmap for network scanning and Wireshark for sniffing. This tool produces the output without any interpretation. Using Scapy, we can create and send custom packets over the network and analyze the raw output received with a minimal amount of lines of code, and it supports a wide range of protocols for the purpose[3].

A. Scapy Features

There are lot of tools which are used to build and interpret the packets in various ways like packet forging, sniffing, testing, scanning, fingerprinting and attacking. The Scapy is one of the best open source tools for the above operations. The major features of Scapy are mentioned below.[3]

- Fast packet designing
- Default values that work
- No special values
- Unlimited combinations
- Probe once, interpret many
- Interactive packet and result manipulation

B. Python Scripting using Scapy

The scapy is a tool written in python language. Now, this section narrates, the scapy is used as a library function in python, to run various packet manipulation functions in terms of IT security. Here, we are using gedit as an editor to writing basic python programs using scapy. [4]

```
# g e d i t s n i f f s c a p y . py from s c apy . a l l imp o r t
p a c k e t = s n i f f ( c o u n t =50) p a c k e t . nsummary ( )
```

Save this program, compile and run as follows:

```
# python s n i f f s c a p y . py
# . / s n i f f s c a p y . py
```

This will sniff for 50 packets, named as P and it will print a summary of the 50 packets that were discovered using sniff command in the above program [4].

```
# g e d i t s e n d r e c e i v e . py
from s c apy . a l l imp o r t ans , unans= s r ( IP
( d s t = " 194 .165. 85 .101 " , t t l =1 5 ) / ICMP ( ) )
ans . nsummary ( ) unans . nsummary ( ) p= s r 1 ( IP
( d s t = " 1 9 2 . 1 6 2 . 8 2 . 9 0 " ) / ICMP ( )
/ "JATAML" )
p . show ( )
```

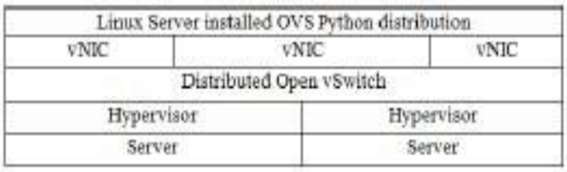
- The answer and unanswered packets, which displays how many packets has answered and unanswered respectively The function sr1() is a variant that only returns on packet that answered the packet that was sent.
- sr() and sr1() are used for layer 3 (Network layer) packets only.
- If you wish to send and receive layer 2 (Data Link Layer) packets, you must use srp() or srp1().

```
# g e d i t commandprompt scapy . py imp o r t s y s
from s c apy . a l l imp o r t s r1 , IP , ICMP p a c k e t = s r 1 (
IP ( d s t = s y s . a rgv [ 1 ] ) / ICMP ( ) ) i f p:
p . show ( )
```

The above program starts system argument as an input. The sys.argv[1] as the destination address states that after executing the script, the first argument to follow the execution of the script will be used for the destination address, for example: #./scapysr.py 192.158.81.120.

III. vSWITCH HARDWARE VIRTUALIZATION

vSwitch is an open source multilayer software switch is developed for network segmentation and traffic visibility. It has a transactional configuration database with Python bindings. It is used to implement a quality switch production platform that encourage standard management interfaces and opens the programing based forwarding functions to control. It throws away the hardware switch with the incorporation and the support of Linux kernel. It supports multiple open source OSs based virtualization technologies such as XenServer and virtualBox [5]. The fig-1 below shows the OvS design support distribution across multiple physical servers implementation.



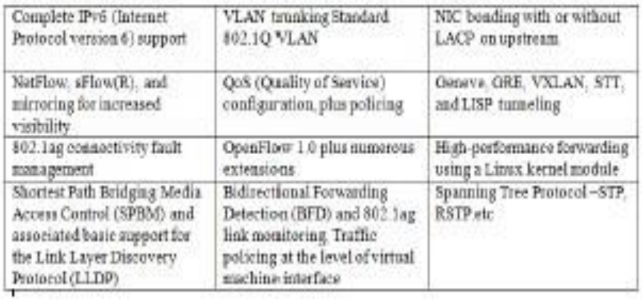
Linux Server installed OVS Python distribution		
vNIC	vNIC	vNIC
Distributed Open vSwitch		
Hypervisor	Hypervisor	
Server	Server	

Fig.1. OVS distribution between multiple physical server

Open vSwitch functions are suitable for virtual machine environment. In addition, it has a standard control and interface visibility to the layer three virtual networking and support across multiple physical servers. The programs ~~are~~ written in python are easily launched to other environments as it supports platform- independent. However, with Python API support are the vSwitch's functionality with Linux kernel support over physical switches, some of the features are discussed in the following Fig.2.

A.Open vSwitch APIs Open

vSwitch operates in the user space doesn't require kernel module. It is an advantage of oSwitch and it is easier to measure performance than kernel based switches. Some of the Python API distribution by the Open vSwitch is defined in the subsequent lines. The main distributions are ovs-vswitchd is used for implementing switch damen, ovsdb-server is function as a lightweight database server, ovs-dpctl is used for configuring kernel module of switch, also support XEN server installation and ovs-vsctl, ovs-appctl, used for running , querying and updating the configuration.



Complete IPv6 (Internet Protocol version 6) support	VLAN trunking Standard 802.1Q VLAN	NIC bonding with or without LACP on upstream
NetFlow, sFlow(R), and mirroring for increased visibility	QoS (Quality of Service) configuration, plus policing	Geneve, GRE, VXLAN, STT, and LISP tunneling
802.1ag connectivity fault management	OpenFlow 1.0 plus numerous extensions	High-performance forwarding using a Linux kernel module
Shortest Path Bridging Media Access Control (SPBM) and associated basic support for the Link Layer Discovery Protocol (LLDP)	Bidirectional Forwarding Detection (BFD) and 802.1ag link monitoring. Traffic policing at the level of virtual machine interface	Spanning Tree Protocol –STP, RSTP etc

Fig. 2. Features of Open vSwitch

B.Popular Open vSwitch tools

Open vSwitch has several tools that provide helps control the network functionalities, some of them are discussed below, ovs-ofctl is a utility used for querying and controlling switch controllers, ovs- pki, a utility used for public-key infrastructure, ovs-testcontroller, a controller for testing, tcpdump used for parsing the OpenFlow messages and finally the qdisc for supporting hierarchical fair service curve method of queuing discipline. This section discusses one of the usages of Open vSwitch traffic rate-limiting and rate bursting implementation methods. It is to rate-limit and QoS policing of traffic sent by a VM controlled by vSwitch which could vary from Kbps to Mbps at time. This experiment starting with one physical network and it consists of Ethernet data network for virtual machine data traffic. This physical network is used to send traffic from and to host and for measuring the rate of traffic sent by each virtual machine. On the other hand, this physical network is considered optional and we connect all VMs to a bridge that rather connected to a physical interface. It uses tools to monitor the host activity.

C. Experimental environment

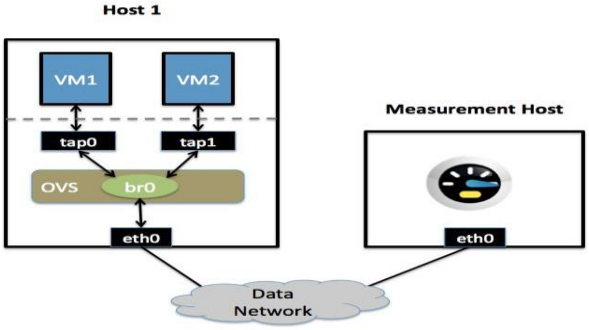


Fig.3.Open vSwitch Environment

As far as the physical hosts are used in the VM environment as shown in the Fig.3, it is named two physical host, named Host1, Host2. Host1 is a terminal that runs on single network interface card together with open vSwitch. Also the physical NIC participation with OVS bridge, eth0, is connected to the Data Network and no IP address would be assigned.Host2 is a monitoring or measurement Host, capable of measuring VM's throughput. Netperf[7] is a tool that tests the rate of traffic from one host to other while travelling, this host with single NIC and a IP address that would enable reachability. There are two Virtual Machines (VMs) namely VM1 and VM2 installed and run in the host1. The configuration of each VMs consist of single Linux device interface on the physical host, these interfaces changes from server to server and host to host, also based on the connection methods, for example, interface name vif1.o for XEN servers.

D. Configuration Steps

The VM1 and VM2 are configured with ingress policing rule based on the configured interface table. It has been set to two values, namely ingress policing_rate and ingress policing_burst. The policing rate values enable the maximum traffic rate in mbps that is allowed to send through VM. The rate policing parameter to indicate the amount of data traffic send by the interface and it shows bursts when more than the rate set by the policing rate exceeded. The following commands are used in hosts for rate limit, burst between VM1 and VM2, For VM1 rate limit and burst are set as shown below,

Ovs interface XEN server vif1.o	VM1	VM2
Ingress QoS rate	100	1000
Ingress QoS burst	10	100

The applied rate and burst limit can be viewd through the following commands ovs-vsctl list interface vif1.o.

E. Testing and troubleshooting

The Python based netperf is used for testing virtual machine rate limiting and burst rate in the host that is used for monitoring and measurement purposes. Also netperf has a clientside and a serverside namely netperf and netserver. We run netserver on the host that monitors the traffic flow rate. Say the host IP address for the measurement purpose is set to 172.16.1.1 and is reachable from both virtual machines terminals. Now we run the command "netperf -H 172.16.1.1" on both VM1 and VM2 with ten to twenty seconds interval and measure the detail throughput values, Both VMs are sent as TCP traffic to the measurement host. Here the speed limit is set to 10, 100 1000 to 10000 bps. In order to view the rate limit progress ingress qdisc should be enabled and verify the userspace /sbin/tc[6].

F. Output Interpretation

The command netperf -H 172.16.1.1 -t {UDP_STREAM/TCP_Stream} -- -m 1000, used for both TCP and UDP stream of data across the VMs using Open vSwitch. These kind of virtual switches used only for rat limiting and not used for queuing the packets. It drops after the traffic rate reaches the threshold point or burst rate.

IV. OTHER NOTEWORTHY FREE AND OPEN SOURCE SOFTWARE

A. GNU Octave Programming

In this section, we address yet free software known as Octave. It is a higher level programming language primarily developed as a major free alternative for highly expensive Matlab programming tool. The Octave tool is essentially used for numerical computations. The emphasis is to use a code that is syntactically compatible to Matlab so that users can find it as free alternative for Matlab tool. Octave can be also used as batch-programming. Furthermore, like Python, Octave programming tool too supports command line interface in addition to script based interpreter. Fig.4 shows Octave tool running on a Linux platform[7].R Programming Of late, R is a popular programming tool and software paradigm for statistical computing and graphics. This programming tool is widely used by statisticians and researchers of data mining, as it helps in providing good support for analysis and visualization of data. This is one of the free software available under GNU package. The R programming has a command line interface and it comes with variety of graphical front-ends. R is supported with comprehensive library to implement various statistical and graphical algorithms that includes linear, non-linear modeling, classification, clustering and popular machine learning algorithms. R libraries could be easily extendible through functions and extensions. The community of R is well known for its active contribution in the development of packages. The standard functions of R, are written in R itself. The C, C++ codes can be linked and called through R during run time. In addition, R objects can be manipulated through C, C++, Java, NET, Python. This R programming was introduced as an alternative to S programming language. Furthermore, R has tools to produce publication-quality graphs and mathematical symbols.

V. CONCLUSION

This paper describes the comparison between of open source tool (python) and proprietary software (Matlab) in the programming environment. The administrator uses the packet manipulation tool like Scapy, written in python language, to manipulate the packets for easy system administration, which is narrated in this paper. The transactional configuration written in Python, is used in Open vSwitch multilayer software switch for network configuration is described in the third section of this paper. Finally, this paper discussed about other open source software and their features, which is useful for the researchers and students to implement their applications in open source platform.

References

[1] “Python Homepage.” www.python.org.
[2] https://en.wikipedia.org/wiki/Scapy.
[3] http://resources.infosecinstitute.com/scapy-all-in- one-networking-tool/
[4] http://www.scmdt.mmu.ac.uk/blossom/downloads/ byDoing/PythonScriptingwithScapyLab.pdf
[5] “Traffic Control, Online Documents.”
[6] “Netperf.”, Netperf, http://www.netperf.org/netperf/ [7] http://octave.org/download.