Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2020, 7(10):121-130**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Evaluation of Loop, Star and Tree Virtual Operating System and Switches for Streaming Wide Datasets

Adegbenjo, A.*, Adekunle, Y.

Department of Computer Science & Information Technology, Babcock University, Ilishan-Remo, Ogun State
*Corresponding Author Email: adegbenjoa@babcock.edu.ng, Phone Number: +2348035859474

Abstract This finding examines the architectural efficiency of virtual operating switches and the networks that manage package deliveries of broad streaming and complex data systems, under different topological frameworks. The tree, mesh, and star are completely related to the topology structures examined. A virtual architecture for network development with re-configurable parameters is created. Local administration of HTTP and SSH maintenance was allowed for each node. This permitted a different configuration of each node, either through a web interface via the HTTP or through an SSH protocol command line. In the early stage of this article, control of the web interface was critical because it makes it easy to display which configurations may be changed and allowed. The user interface makes system improvements for the web browser only. The finding has an SSH GUI far more essential since it enables commands on a command line to be transmitted to a node. When a web interface will only alter by clicking on the button, it can only send commands through the clave via the command-line interface. This allows it even faster to alter a huge number of nodes, as with a few keyboards commands several improvements can be made. Per node needs to be configured separately, and SSH management allows for this role, by having a network of several nodes.

An estimate of memory, disk space, and Processor use on three separate virtual network switches were also examined (open switch, open switch, openWRT). The Bandwidth Specifications are checked for two particular protocols – Routing Information Protocol (RIP) and Free Shortest Route First (OSPF).

Keywords computer networks; performance architecture; open switch

Introduction

A physical device would be more useful for building a broad test network, so it would imitate a layout of actual living. However, a physical device will be incredibly time-consuming and very costly for a wide network. A virtual machine was then selected to simplify network development and setup. The virtual machine host program is classified as a hypervisor. In this article, Oracle VM VirtualBox and VMware vSphere ESXi, two hypervisors have been included. Oracle VM VirtualBox is a free open source hypervisor powered by numerous operating systems which includes the assistance of the virtual machines for testing and is herein named VirtualBox. VirtualBox was the most common edition of this document, because of its free-to-use policies and adoption of all the major operating systems[1]. However, VMware vSphere ESXi has been used on a high-performance machine built for [2] owing to the widespread usage of the document as a source of energy.

There were three operating systems for the virtual switch: Free vSwitch, OpenSwitch, and OpenWRT. These operating systems have been picked because they are open source, free to access and they have a wide market share. An open-source helps a person to alter and develop the system according to their wishes. This will help to remove redundant programs and other space-saving data. A payment for device upgrades is expected for certain virtual operating systems that run on the market. It may be used without any purchase or subscription to the OS without using the operating system. Because of their very wide commercial use, the key justification for preferring these switch operating systems was its. Being one of the most common operating systems contributes to more regular upgrades, security fixes, and a wide audience on the Internet. The three operating systems and their architecture are quite close. Anyone may be mounted on a physical company router or network switch directly. They each run their own tweaked Linux edition, which is optimized to accomplish the same objectives for numerous applications. Since all the operating systems are focused on Linux flavors, the operating system may have various hardware characteristics to operate as expected. The file size and memory constraints are the main contributors to the hardware specifications.

Literature Review

[4]. Open vSwitch is special since it is a toolset rather than an entire operating system. Open vSwitch launches most Unix and Linux related operating systems for download. This renders Open vSwitch valuable since it can be mounted on most equipped devices. Open vSwitch offers a picture of the current iteration of the tools as well as a Linux operating system Virtual Machine Appliance. Open vSwitch is special since it is not designed to be mounted like many of the other available operating systems on physical switches. For the following experiments, the simulated picture was used.

OpenWrt was the picked operating system for the final virtual transition. It's an embedded Linux-based operating system. OpenWrt was first written in 2004 and is meant for Linksys routers as alternate firmware. [5] Emphasis on being a built-in OS implies optimizing the firmware for low resource use as the size, Processor, and memory. Since OpenWrt has a bare minimum flavor of Linux, is the least versatile virtual switch operating system picked. The method for OpenWrt updates includes the most recent release to be compiled and the firmware reflashed on the system. From a virtual operating system viewpoint, the virtual machine has to be re-installed in the hypervisor for each upgrade and the settings change. This transition is the most time-consuming transition in the method and the OpenWrt digital machine was also not upgraded during this finding to save time for the new releases. Every virtual operating switch device was similarly designed for this article. Many common choices, such as default user names and lack of passwords were left to make it easy for users to obey user guides and so this finding did not influence the aim of developing a virtual network adversely.

Methodology

There were two methods of defining the network layout. First, you build several network nodes and change them for any form of structure selected while running. The second is to construct the built framework by node modules in Figure 1 using a single node and cloning multiple times.

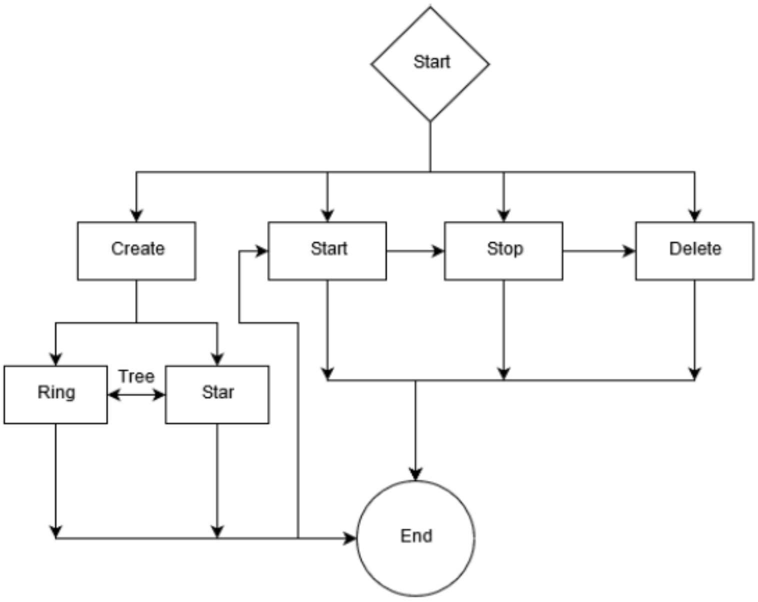

Figure 1: Node Modules in Architecture Hierarchy

Each way has its benefits and drawbacks. If each node is attached to the ideal network layout and each node has a downside, the first approach to minimize time is to upgrade or re-create a disadvantage if the operating system improves. The first method uses the opposite pros and cons. The second form by utilizing a common source node. A pro that only one node must be upgraded or changed, but a pro that any node-set must be upgraded in the network is a problem. In the end, the second option was preferred because it only changed the root node and permitted the operating system to be tweaked or upgraded. This spares substantial time as the whole system has to be modified. The phase of network development should begin with a single cloned node. This node is sometimes named the 'source code.' The root node is a hyper view node that is pre-configured and imported. Any configuration made in the main node is held in all child nodes unless they are changed individually. Using one node for the main network structure and cloning it, the whole framework is configured for each selected network structure from the ground up. A way to automate the creation of nodes in VMs has been created to set up a large number of nodes needed in the framework. There are two methods nodes in hypervisors may be cloned and installed. The hypervisor provides the graphical user interface and a command-line interface, including the setup access to the virtual network switches. The interface of a command line enables the formation and modification of nodes in the graphical user interface that require several clicks. The GUI on the control line has been selected as this is the simplest way to automate device development.

The configuration was written in python. The Automation script was configured for four major machine functions: Developing, Beginning, Halting, and Removing. The most complex is the creative function of the script. The first hardcoded in the script is the right insight on how the device is to be set up. These configurations include the number of device nodes, which clone node, the network structure form to construct, and other choices.

The roles beginning, halting, and removing Simple. Simple. These functions just take the amount of Project nodes as feedback to complete. The beginning feature Starts a virtual machine one by one to test if Hypervisor has ample capital to launch a new node. If the hypervisor is low, the script is low Avoid stopping the hypervisor from locking sooner Any restart of the whole machine will otherwise be required. The limit to the limit Only avoids or deletes the specified feature Delete Number of nodes and number of domains.

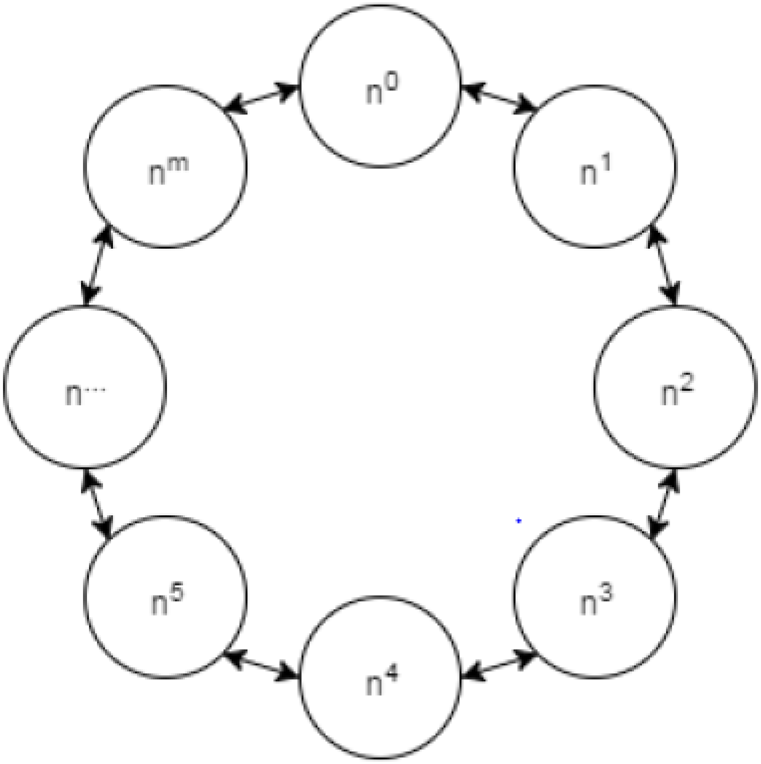

Figure 2: Ring Topology

Three networks for the node-link method Automated topologies have been programmed. The selected Topology was first developed in a data structure form array Later it was transcribed to the phase of development. The 1st one He's a topology ring. Per node has a ring topology. With two single-chain links. The network of networks packets on which a ring topology is destined between two selected nodes as seen in the figure each node 2. Ring topology was automated with the implementation of m max nodes connecting the n node to and the node where the final node *m* was connected to n^{n+1} and n^{n-1} node where the final node *m* was connected to the nodes n^1 and n^{m-1}

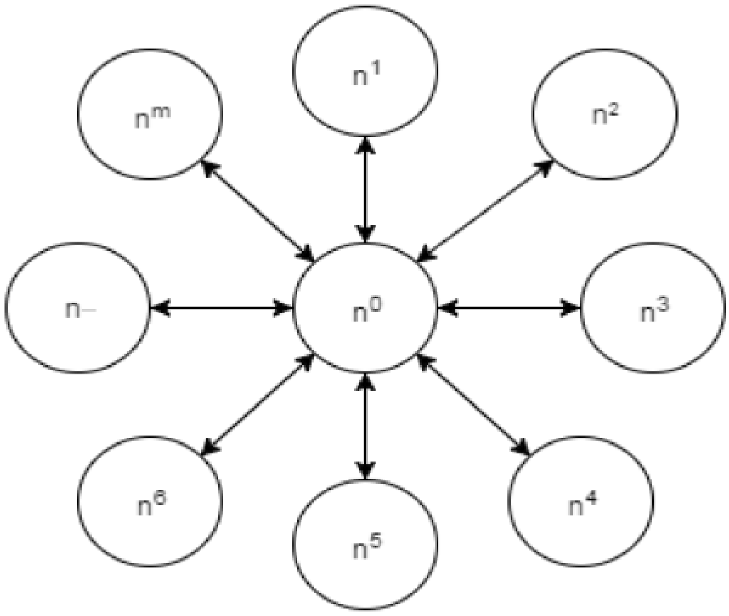

Figure 3: Star Topology

Three networks for the node-link method Automated topologies have been programmed. The selected Topology was first developed in a data structure form array Later it was transcribed to the phase of development. The 1st one He's a topology ring. Per node has a ring topology. With two single-chain links. The network of network packets on which a ring topology is destined between two selected nodes as seen in the figure each node2. Ring topology was automated with the implementation of m max nodes connecting the n node

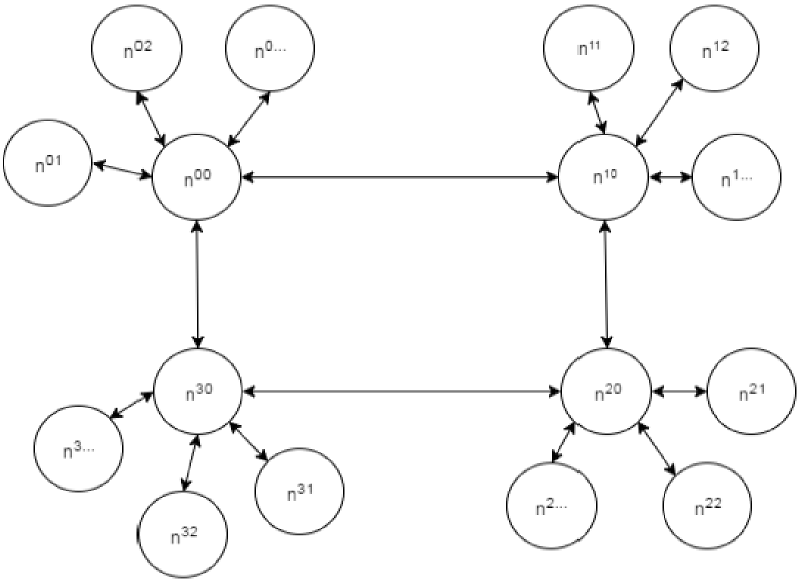

Figure 4: Tree Topology

A ring mixture and the topology of the tree. The topology of Star. Star. Automation was first done by Ring topology, then the node list to the star Feature of topology to connect any node in the ring The estimates in Figure 4. While the maximum cap is the same number of nodes per star, an infinite number The ring will link several star networks.

Network Routing-Types and Protocols

There are two major routing choices for static and dynamic traffic packets in computer networking. The manual way to construct routing tables is through static routing. This helps a network manager to determine specifically where the goal node is to access the data packets. The routing tables must be modified if the network shifts by utilizing static routing. Static routing is therefore not suitable for dynamic, extremely vast, and constantly evolving networks [7]. Dynamic routing provides benefits to solve the challenges of static routing. Dynamic routing ensures that a routing table centered on a routing protocol is automatically generated. Dynamic routing protocols will then control and document network adjustments. Change the tables accordingly if required. if required. Policy dependent routing (PBR) is used with complex routing protocols in decision making. The policies include packet size, the form of protocol, ride duration, bandwidth, and trust. There are two major subforms within complex routing protocols as to how to carry out the routing table upgrade process. The first was the protocols for link-state routing. Each node in the network is used by the connection state protocol to construct a diagram. Each node on the network then uses this chart for the best possible route to the destination of each packet independently. The network as a whole must be mapped and sent to each specific node to complete this goal. This may have a huge effect where a wide quantity of nodes is found in the network [8]. The second type of protocol for dynamic routing is the protocol for routing vectors. A node doesn't know the direction along which data is routed in this way. Rather, nodes focus their routing on the path and distance of a packet. When upgrading the routing tables, this protocol is more effective than the link-state protocol, since nodes can only map a distance for one node to hop away. However, the distance vector routing protocol cost a lot, since the node doesn't know the full network route when sending packets. A packet is sent to several nodes, resulting in increased network latency and use [7]. In comparison, there are many such routing systems. This compares with how data are distributed. These technologies include unicasting, streaming, multicasting, whatever cast, and geocaching. Unicast sends data to one specified node, sends information to a node in the method, transmits node to a specified node community, sends data to a node group based on a given policy, and sends data to a geographical region. Route protocols include the RIP, the IS-IS, Free Shortest Path First (OSPF), and Boundary Gateway Protocol (BGP). The most popular routing protocols include RIP. One of the oldest protocols for routing is RIP. RIP is the routing protocol for the distance vector used to upgrade the nodes with transmitted or multimedia. RIP is one of the most commonly used protocols because no set-up step is necessary. Using IS-IS to upgrade nodes using multicast. For routing, IS-IS does not use IP. OSPF is another multicast-like link-state protocol, IS-IS. Unlike IS-IS, though, the Internet Protocol (IP) is used for OSPF. OSPF is special in that it utilizes its IP number to relay information rather than a transport protocol. BGP is a standard for the external gateway or Boundary Gateway Standard. This kind of protocol is used to link various internet networks [8]. Four network topologies have been used to evaluate bandwidth constraints: node, ring, mesh, and a mixed node and ring configuration. To connect in the network topology, two virtual test machines have been used. Due to the usage of open source and its familiarity with the applications running on the OpenWrt node [9], the operating system Ubuntu, a Linux-based operating system, was selected to be built on the virtual test machines for this work [1]. The Ubuntu fragrance is version 16.0 of Ubuntu 32x. Owing to its reduced disk space specifications and lower usage of random access memory compared with the 64-bit version, a 32-bit operating system was selected. The new stable update of the Ubuntu Software family updates 16.04 which currently provides technical support for all previous releases. Through installing the operating system on one virtual machine, both Ubuntu virtual machines (VMs) were developed and then the machine cloned. One VM and the other the destination computer is selected. The root VM starts contact across the network to the goal virtual machine.

Ubuntu A is a source VM, whereas Ubuntu B is the destination VM. Every virtual machine has almost been attached at a default position in the topology of the node network and may be placed at a later time varied. By using the Quality of Service (QoS), network bandwidth can be individually managed by network switch nodes. Service quality settings enable a network administrator to restrict each switch's bandwidth and to prioritize those packets or packet sources. The link to a switch may also be improved by utilizing Quality of Service to enable a certain bandwidth cap to cross. This helps the various routing methods evaluated for this work to assign some importance to the usage of nodes. Multiple programs used to relay the data from Ubuntu A to Ubuntu B and back for checking congestion, VM links. As the primary test program, Iperf was used. Iperf is a platform primarily designed for bandwidth testing [10]. For this job, the program has been chosen because it includes many criteria for the data form to be streamed. Such criteria include the packet size definition and random or input data streaming. Iperf is used as a server and/or client setup in which each computer can explicitly calculate bandwidth using network topology. For benchmarking this job the default Iperf test settings have been used. This involves the usage of the 5001 TCP port with 85.3 kbytes of window size [11]. The Iperf produced random streaming data to be sent between server and device. The average bandwidth consumption was calculated by a device of three or four Iperf checks. The virtual network's latency was calculated using the 'ping' control line function. Ping sends an ICMP packet format to the target computer and receives an "echo" packet from the goal that contains a link log. The ping program analyzes the various echoes and calculates the network failure of the packet and the minimum, average, and maximum round trip latency. The default ping settings for this job are a 32-byte parcel, a packet timeout for one second, and a "loss" [12]. Tests could be done for 20 counts until the average device latency for the round trips was identified.

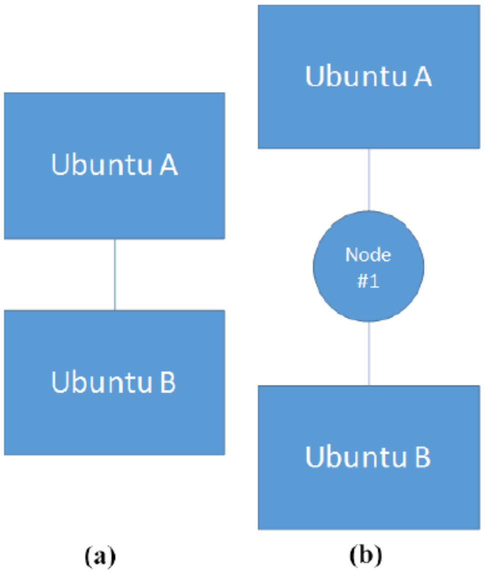

Figure 5: (a) Benchmark 1, (b) Benchmark 2

This was required to define the full networkability before adding them to network topology. This was the first phase in the network to explicitly connect the device to the server Ubuntu A with Ubuntu B. The network bandwidth is restricted by the hypervisor and the CPU consumption when two virtual machines are connected directly. The overall performance is approximately 2.5 Gbps as can be seen in Figure 5a. The average latency between the two virtual machines, which is 0.205 ms, may be found to be technically unlikely to surpass the bandwidth and latency determined from a direct link from either of the network topology examined in this experiment. To calculate output reduction by one turn, the second requirement for this method was to link all simulated machine with the intermediate node. The Ubuntu A was linked to an external network interface through a network node (e.g. see Figure 5b-node # 1) while Ubuntu B was linked to an interface inside the switch node. A static routing system has now been designed to use the switch node to allow for minimal decision-making for all traffic. The system's peak throughput was about 2.5 Gbps, almost the same as the first benchmark, and was then again used to calculate the bandwidth from Ubuntu A to Ubuntu B. The research was then performed and the average latency was found to be 0.321 ms, which revealed that the average network bandwidth is not influenced significantly by having fewer multiple nodes in the device. However, the findings of the latency test demonstrate that for any switch traversed in network topology, an average of 0,116 ms of latency is obtained. Two requirements prove that the device moves data at an unbelievably high pace. The typical physical network latency is between 1ms and 4ms by one turn. However, by utilizing good standard appliances and cable technology, these rates are not surprising. These parameters mean that this virtual network should simulate a real network.

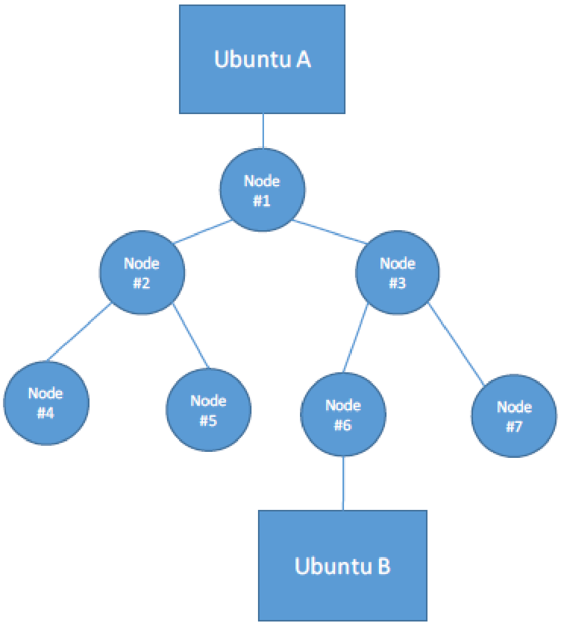

Figure 6: Ubuntu"s in Tree Topology

A topology tree in Figure 6 was the first network checked. The topology of the tree was generated by the use of seven virtual switch nodes through the link of two other tree topological nodes through the connection of two other nodes with this node. To use the RIP protocol, each switch has been set up. This routing protocol establishes a routing table permanently by sending a set of hops to all routers. Under the sixth node in the 3rd level of the tree, the virtual server Ubuntu B was linked. Ubuntu A was attached to the virtual client on the tree level above the first node. A packet must cross the nodes in at least three hops while transmitting from Ubuntu A to Ubuntu B.

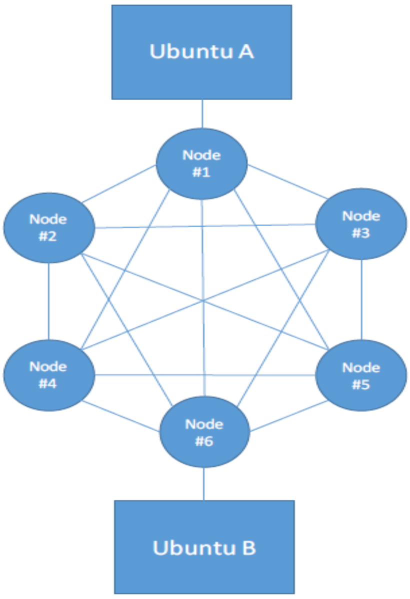

Figure 7: Ubuntu"s in Fully connected Mesh Topology

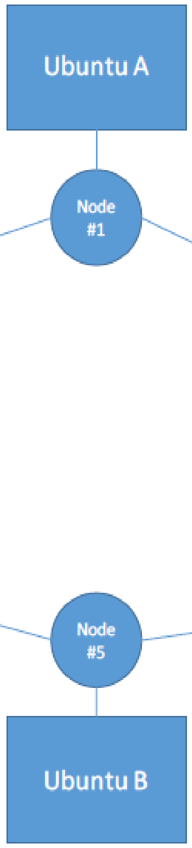

Figure 8: Ubuntu's in Ring Topology

The second topology examined as seen in Figure 8. The ring topology was defined by the combination of eight virtual switch nodes with two other nodes at first. In a clockwise manner, the nodes are numbered. In the same number of hops, the virtual client and server machines were linked together. A completely linked mesh topology was the third topology studied. This topology was generated by attaching each node to every other node in the structure, as seen in Figure 7. The nodes of the client and server were both similarly remote

Results & Discussions

Open Switch versus Openwrt Performance with Memory, CPU, and Disk space

The VMs were tested using the High-Performance Computing (HPC) cluster of the University of North Dakota for scalability. Tests also have shown that during the initial start-up process of a node development a significant amount of Processor use is consumed. A node can run an average of 5% CPU ability of the test platform. To address this issue, the program, before starting the next node, has been changed to allow the nodes to be started for five seconds. Although this delay period fits fine for a smaller system of nodes, it will take potentially about 80 minutes to complete the same delay on a thousand nodes system.

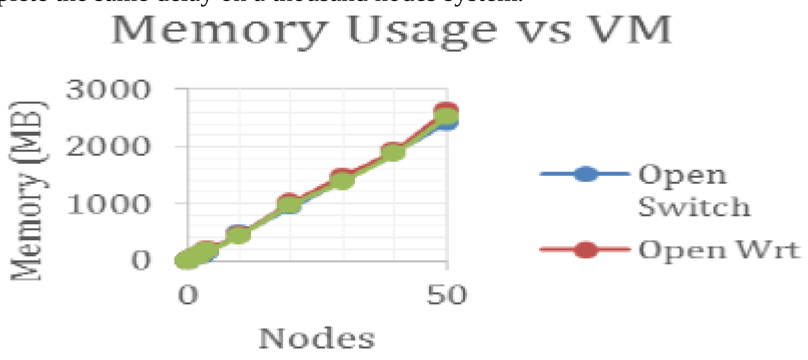

Figure 9: Memory versus VMs

The memory and Processor usages of the machine were reported to find the best numerical nodes as shown in Figure 9 with several nodes and a mixture of different topology. The diagram displays the CPU by the operating system in Figure 10. If the number of nodes grew, Open Wrt and Open vSwitch increased relatively slowly with their usage of the Kernel and Open Switch increased abnormally large use of the Kernel.

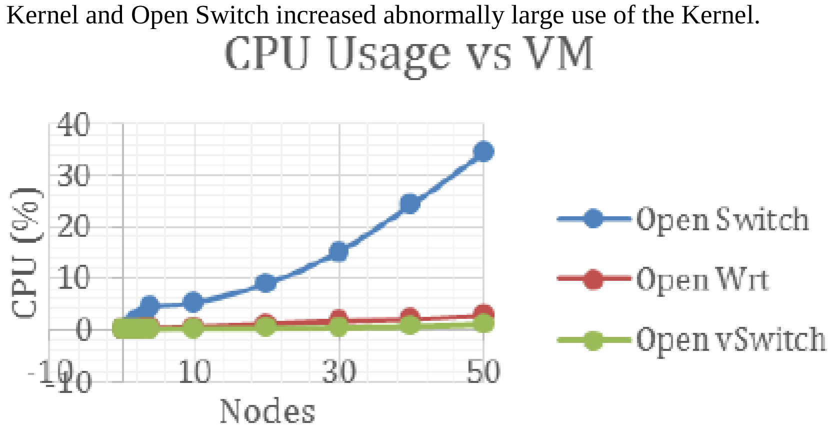

Figure 10: Memory versus VMs

The use of memory per node was linearly increased for all running resources, the machine slowed drastically and caused the machine to almost freeze. Although memory use was the only aspect that restricted the creation of a wider network, none of the tested operating systems had any gain. The disk space consumption as seen in Figure 11 is one of the main differences between virtual operating systems. Open Wrt requires 11 Mb, Open Switch 500 Mb, and Open vSwitch 1,320 MB to be used by one node. The utilization of storage space is a consideration in the selection of a virtual machine. The sum that Open vSwitch can't reach its current setup is 200 GB due to the HPC allocation to students.

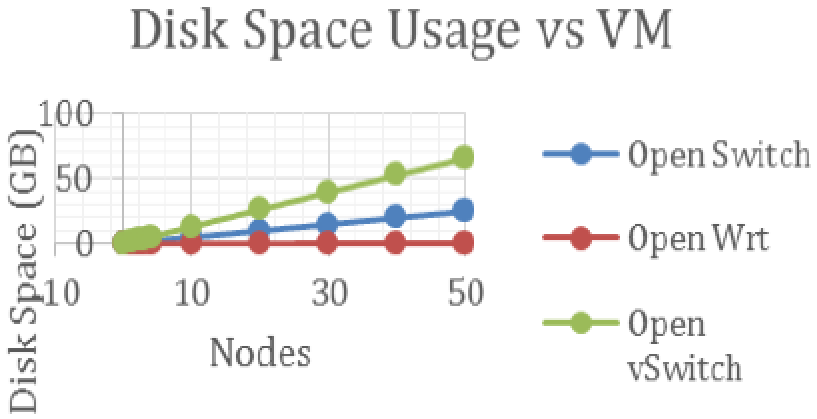

Figure 11: Disk Space Usage vs VM

Conclusion

An automated virtual architecture under the tree, mesh, and ring topology is investigated against CPU, Bandwidth, and memory requirements. For scalability, the VMs were tested using the University of North Dakota‟s High-Performance Computing (HPC) cluster. The piloted program proved a success for streaming data sets and was immensely helpful creating many nodes within a short period. Remote access through the wide-area network interface could be disabled on each node to prevent an outside source from attempting to gain access and modify the configurations. Additional features and options however will consume more memory and need to be weighted for their usefulness through trade-off analysis.

Reference

[1]. "Virtualization Oracle", Oracle.com, 2016. [Online]. Available: https:/www.oracle.com/virtualization/index.html. [Accessed: 17- Dec- 2016].

[2]. J. Wickboldt, R. Esteves, M. de Carvalho and L. Granville, "Resource control in IaaS cloud systems made scalable by programmability", Computer Networks, vol. 68, pp. 54-70, 2014. "vSphere ESXi Bare-Metal Hypervisor", Vmware.com, 2016. [Online]. Available: http:/www.vmware.com/products/esxiand-esx.html. [Accessed: 17- Dec- 2016].

[3]. "OpenSwitch", Openswitch.net, 2016. [Online]. Offered: http:/www.openswitch.net/. [Installed: Dec 17, 2016].

[4]. Open vSwitch, Opera.org, 2016. Opera. [Online] [Online]. Different options: http:/openvswitch.org/. [Click here: Dec 17, 2016].

[5]. "OpenWrt [OpenWrt Wiki] Edition History," Wiki.openwrt.org, 2016. [Online]. Online]. Display: http:/wiki.openwrt.org/over/history. [Accessed: Dec 17, 2016]. [Accessed:

[6]. J. R. Esteves, M., Wickboldt. The L. and de Carvalho. Granvilla, Data Networks, vol. "IaaS resource management made programming scalable. 68, 54-70, 2014. 2014.

[7]. Columbus, Vassos. "RIP, OSPF, IS-IS and EIGRP Routing Protocol Compare of Results." 2015. 2015.

[8]. R. A. and Graziani. Johnson, Procedure and Principle Routing, first ed. Cisco, 2007. Indianapolis, Ind.

[9]. 'Ubuntu | Ubuntu' [11]. Álvarova.com. Remember, 2017. 2017. Website. Web. The 9th of May 2017.

[10]. WELCOME, Vivien. "Iperf-Iperf3 and User Recording Iperf2." Iperf.fr.-Iperf.fr. Note: 2017. 2017. Website. Web. The 9th of May 2017.

[11]. WELCOME, WELCAL. "Iperf-the bandwidth measurement tool for TCP , UDP and SCTP" Iperf.fr.-Iperf.fr. Note: 2017. 2017. Website. Web. The 9th of May 2017.

[12]. "Linux Man page" (Ping(8)). the.net. the. Note: 2017. 2017. Website. Web. The 9th of May 2017.