2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems

Performance Evaluation and Analysis of the Virtual Networks Built by Different Virtual Machine Managers

Zhongjiang Yao[1,2] yaozhongjiang@iie.ac.cn
Xiaodan Zhang*[,2] zhangxiaodan@iie.ac.cn
Yuepeng E[2] eyuepeng@iie.ac.cn
Tong Li[2] litong@iie.ac.cn
Jingguo Ge[2] gejingguo@iie.ac.cn
Haina Tang[1] hntang@ucas.ac.cn

[1] University of the Chinese Academy of Sciences, Beijing, 100049, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100195, China
*Correspondence author

***Abstract*—Virtualization is a key technology in research fields such as cloud computing, big data and new networks. A lot of cloud computing platforms, data centers, intranets and test environments of networks developed in recent years all depend on virtualization. Virtualization technology has become a hot research focus in computer industry because of its flexibility and wide application. By analyzing the data communication of different virtual machine managers (VMMs) for open source virtual machines (VMs), the study compares the performances of the virtual networks separately supported by these VMMs in view of some indexes. These indexes include Throughput, Jitter, Packet Lost and RTT. By doing so, it provides a valuable reference for the choice of virtual technologies for virtual networks.**

***Keywords*—virtualization; test environment of networks; performance evaluation; KVM; Xen; XenServer**

I. INTRODUCTION

Virtualization is an important technology in cloud computing, big data and network testbed. Various physical components of a computer, including the server, network, memory and storage, are abstracted, converted and presented by employing the virtualization technology. By doing so, the barrier that the entities are inseparable is broken down, so that users can take advantage of these components in a better way compared with the original configuration [1]. Using the virtualization technology can solve the drawbacks of existing networks like inflexibility and closure to conveniently offer computing and storage capacity. In addition to this, it also can improve the utilization rate of resources and reduce the energy consumption because it distributes the physical resources according to one's needs.

Virtualization has developed nearly half a century since its start. However, its advantages haven't given full play due to the constraint of traditional network architectures. In recent years, more and more problems of traditional network architectures have emerged with the rapid change of the network traffic and application requirements. To solve these problems, the companies, universities and research institutions of numerous countries in Europe, America and Asia set to research and develop future networks. The typical architectures for future networks include software defined network/network function virtualization (SDN/NFV) [2], MobilityFirst [3], CCN/NDN [4] and XIA [5]. Among these network architectures, the rapid developments of SDN and NFV offer an opportunity to the virtualization technology. SDN separates the control plane and the data plane by using the virtualization technology. In addition, it connects several virtual networks together and flexibly controls the network traffic. While NFV migrates various services from a dedicated hardware to a virtual environment by utilizing the virtualization technology. Moreover, these services can be conveniently arranged on a general hardware to achieve various, dedicated network functions. To perform the experiment of the future network architecture, an experimental platform has to be set up by employing the virtualization technology. The experimental platform is expected to be able to organize various networks flexibly and change the network structure at random according to needs. The typical representatives include GENI developed in the USA [6], FIRE built by EU [7] and Sea-cloud Innovative Experimental Environment constructed under the leadership of Chinese Academy of Sciences [8]. Therefore, the virtualization technology is supposed to play an increasingly important role in the network in the future.

The various benefits and promising prospects of virtualization have attracted numerous internet companies to participate in the development of their own virtualization technology. Moreover, the virtualization technology is

978-1-5090-4297-5/16 $31.00 © 2016 IEEE
DOI 10.1109/HPCC-SmartCity-DSS.2016.151

applied to various fields, including intranets, data centers, testbeds, cloud computing and big data. Various projects such as VMware, open source Xen-Project, Citrix [19] and Microsoft all launch different virtualization technologies. The virtual machine managers (VMMs) designed by different developers employ dissimilar methods and show different abstract degrees for hardware, leading to disparity in the performances of the VMs. What distinction of the methods causes the difference of the performance? For this question, lots research groups have carried out many researches and published papers relating the computer performance in the past years. The analysis directions of those papers include scheduling of CPU [9], the abstract degree for hardware resources [10] and the density of VMs [11]. However, in the context that the research on new networks becomes increasingly extensive and important, the performance evaluation of virtual network is scarcely reported. As a result, studying and evaluating the performance of virtual networks and analyzing the influence of virtualization on network performance have vital effects on the development of the virtualization technology and the performance optimization of virtual networks.

To further explore the major factors influencing the performance of virtual networks in the virtualization technology, the simplest virtual networks are set up on different VMMs in this article to evaluate the performance of the virtual networks. First, in the case of same load, the performances of the virtual networks supported by KVM, Xen-Project and XenServer and the physical networks under communication modes UDP and TCP are measured, respectively. Then the performance indexes of the virtual networks supported by (kernel-based virtual machine) KVM, Xen-Project and XenServer are compared under different loads, respectively. The performance disparities between the virtual networks and the physical network are observed, as well as the distinctions of evaluation results among the virtual networks controlled by different VMMs. The authors also analyze the reason leading to this disparity according to the experimental result, providing effective references for the performance optimization of the network virtualization in the future.

The rest of this article is organized as follows: Section 2 introduces the evaluation and research achievements concerning the influence of virtualization technology on the performance of VMs and virtual networks. Section 3 simply analyzes the architectures and the message processing procedures of different VMMs for the virtualization. Section 4 presents the experimental platform and scenarios. Section 5 discusses the experimental data from the perspective of UDP and TCP, respectively. Section 6 summarizes the former experimental research.

II. RELATED WORK

F. Gomez-Folgar et al. published a paper on the Annual IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing (CCGrid) in 2015 [10]. In the paper, they proposed that system compatibility, virtualization software and so on cause some limitation on the properties of CPU and influenced the performance of the applications of VMs as well. They thought that different KVMs and KVM configuration modes show varying abstract degrees to the underlying CPU by analyzing the KVM architecture. In order to confirm the point of view, the performances of the VMs based on KVM that they developed on Encalyptus, Apache CloudStack and OpenStack were tested by using Intel Linpack benchmark, video transcoder HandBrake and 3D DD-DG. The result confirmed the correctness of their view.

Zongjian He et al. analyzed the factors affecting the performance of network virtualization in the cloud computing [12]. They adopted the performance of OpenvSwitch as the major factor for evaluating security and QoS and XenServer as the experimental platform. The influences of OpenvSwitch on the VLAN, QoS, and overload performance were studied by changing the version of OpenvSwitch. The result reveals that OpenvSwitch exhibits good traffic isolation performance and the fine-grained QoS can bring more convenience for the gateways.

Khurana S, Marwah K et al. considered that the scheduling policies of VMs significantly influence the performance of networks [13]. A good scheduling policy can not only improve the resource utilization, but also maximize the economic benefits. They proved that different choices of scheduling policies of VMs exert vital effects on the performance of upper-layer applications and the utilization rate of underlying resources. In order to improve the QoS of cloud computing, they also discussed the performances of VMs applying different scheduling policies.

Current researches on the performance optimization of the VMs mainly focus on the CPU, vSwitch and more abstract scheduling level of VMs. These researches are conducted on the premise of ensuring the running smoothness of a single VM to make VMs occupy resources during communication like physical machines as far as possible. However, other possible factors affecting the network performance in the VMMs are not considered. These factors include the integration methods of management tools, the optimization during data forwarding and the high voltage network. The article attempts to provide a new method for optimizing the virtualization technology by comparing the performance of virtual networks built by KVM, Xen-Project and XenServer, respectively.

III. VIRTUAL MACHINE MANAGEMENT PLATFORM ANALYSIS

KVM, Xen-Project, and XenServer are popular three VMMs for virtualization at present. The application range of KVM is increasingly enlarged and develops rapidly because of its favorable compatibility with the operating system Linux. Xen-Project and XenServer are very mature VMMs, based on which popular software such as OpenStack [18] at present is developed. The performances of virtual networks and networks composed of physical machines have certain differences, while the test result of each VMM is also diverse. The overall architectures of the three open source software

are analyzed in the following to facilitate the understanding of the subsequent experiments.

A. KVM

The architecture of KVM is shown as Fig.1. It can be seen that, to communicate with external servers, the VM need to send or receive I/O requests and then receive responses first. These I/O request messages are issued to the Linux kernel after being driven by the network card of the VM first. After being captured by the I/O Trap Code in the KVM module, these messages are processed and then put into the I/O shared page to inform QEMU [20]. The QEMU receives the specific information of the requests and schedules the hardware to finish the I/O operation. The acquired result is put into the I/O shared page and the KVM module is informed to capture the codes. The read results are returned to the VM [14].

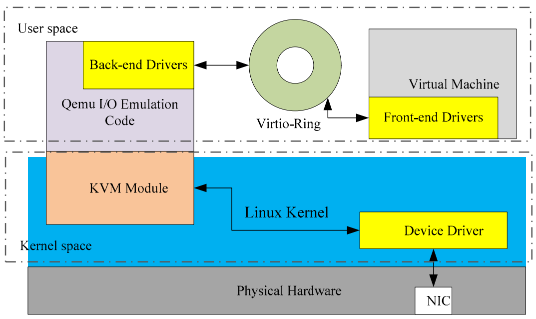

Fig.1. The communication architecture of the KVM virtual machine

The KVM VM is a process of the operating system of the host machine. In the whole I/O process of the KVM VM, the KVM module runs under the kernel mode of the root mode, while the simulation process of the QEMU and the usage of the I/O shared page are performed in the user mode.

B. Xen-Project

The overall framework of the equipment with Xen-hypervisor is presented in Fig.2. To communicate with external servers, the VM has to employ the privileged VM Dom-0 to forward the message and calls the real driver interface in the Dom-0 to call the hardware resources indirectly. When the VM sends I/O requests to external servers, the front-end driver, namely, the driver of the VM is called. Then the request message is put into the I/O shared ring (an area of shared memory) while the data need to be sent are placed in the Grant Table which is accessible for the back-end with authorization. Here, the Grant Table refers to a memory interactively accessed by several Doms [15]. Then, the front-end driver sends the VMEXIT to inform the back-end Dom-0 through the event channel, and the Dom-0 checks and deals with the I/O request during the operation. Afterwards, the response of the Dom-0 is recorded into the I/O shared ring, and then taken out and processed when corresponding VM occupies the CPU [16] [17].

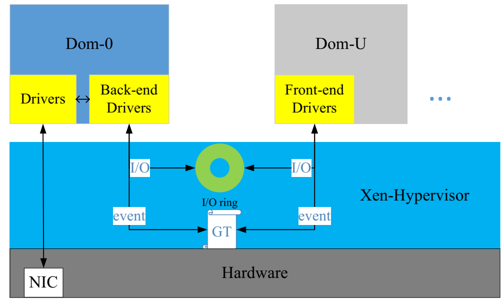

Fig.2. The communication architecture of the Xen-Project virtual machine

C. XenServer

The XenServer is a new architecture developed from the open source Xen-hypervisor with the addition of the modified Dom-0 operating system and kernel and the management tools designed by Citrix. The specific pattern is shown in Fig.3.

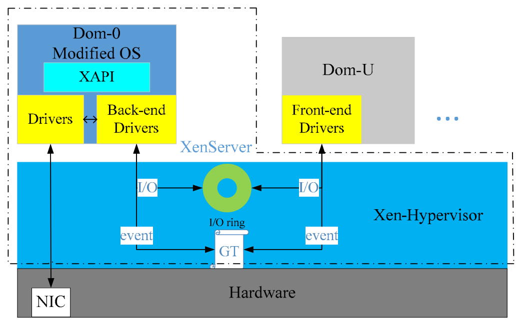

Fig.3. The communication architecture of the XenServer virtual machine

In the whole XenServer, the Xen-hypervisor is still applied as the underlying architecture. Unlike using Xen-hypervisor alone, XenServer is equipped with the customized Dom-0 kernel and system and added with some management tool developed by the Citrix. In addition, Citrix also shields part of information. Therefore, some related information of the process such as the QEMU cannot be observed in the Dom-0 so that some unnecessary functions in the operating system of the Dom-0 are eliminated. In this way, the network performance of the VM improves.

IV. THE EXPERIMENT DESIGN

A. Experimental platform and parameters

The DELL PowerEdge R730 physical machine was applied to conduct the experiment. The operating system used by the physical machine, KVM and Xen-Project host machines was the stable release of ubuntu14.04 LTS, while the stable release of Ubuntu most widely used by servers was adopted as the operating system of the VM. The most widely used release 6.2 was selected for XenServer. In order to more intuitively explain the differences of the network performance of the VMs supported by different VMMs, the information of the software used is presented in Table I.

TABLE I. Software Parameters of the Test Environment

VMM / Specific parameters	Linux	KVM	Xen-Project	XenServer
Version	——	Qemu-img 2.0.0	Xen-hypervisor-4.4-amd64	6.2.0-70446c
Privileged domain OS	Ubuntu14.04.4 LTS	Ubuntu14.04.4 LTS	Ubuntu14.04.4 LTS	CentOS 5.5
Privileged domain OS kernel	Kernel-3.19.1	Kernel-3.19.1	Kernel-3.19.1	Kernel-2.6.32.43
Dom-U OS	——	Ubuntu12.04.2 LTS	Ubuntu12.04.2 LTS	Ubuntu12.04.2 LTS
Dom-U OS kernel	——	Kernel-3.5.0-23-generic	Kernel-3.5.0-23-generic	Kernel-3.5.0-23-generic
Virtual Switch	——	OVS-2.4.0	OVS-2.4.0	OVS-1.4.6
CPU scheduling strategy	CFS	CFS	Credit	Credit

The parameters of the hardware equipment are shown in Table II.

TABLE II. Hardware Parameters Table

Hardware Parameters	
Device models	**DELL PowerEdge R730**
CPU	**Intel(R) XEON(R) E5-2609 v3 1.9GHZ 12 kernels**
memory	**16GB 1600MHZ ECC DDR4**
disk	**8TB**
NIC	**Speed: 1000Mb/s**

The parameters of the virtualized hardware resources assigned to each VM in the experiment are listed in Table III.

TABLE III. Virtual Machine Configuration Parameters Table

Virtual Machine Configuration Parameters Table	
VCPU	**1**
Virtual memory	**512M**
Virtual disk	**8GB**

The tools of the experiment are shown in Table IV.

TABLE IV. Summary of Test Tools

Test Protocol	**Test Index**	**Test tools**	**Correlation Parameter**	**Version**
UDP	Throughput	Iperf	b:set bandwidth of the UDP mode i:set interval of report t:set the sum of test time	2.0.5
	Jitter	Iperf		2.0.5
	Packet Lost	Iperf		2.0.5
	RTT	Ping	i: set interval of report c: set the number of packets	-
TCP	Throughput	Iperf	i: set interval of report t: set the sum of test time	2.0.5
	RTT	Ping	i: set interval of report c: set the number of packets	-

B. Network environment

For all the VMMs, the tests were conducted under the same environment of network topology to ensure the comparability of the test results. The specific topology of the virtual network used in the experiment was introduced in detail in the following.

1. The test result of two directly connected physical machines was used as the network performance reference. The corresponding network topology for test is presented in Fig.4.

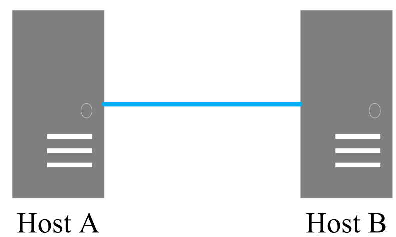

Fig.4. Physical network topology

2. The test result of two directly connected physical machines was used as the network performance reference. The corresponding network topology for test is presented in Fig.5.

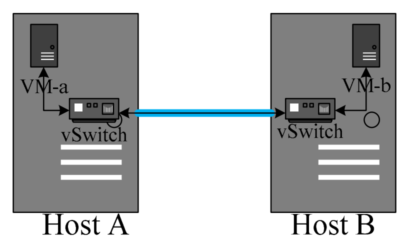

Fig.5. Virtual network topology

V. EXPERIMENTS AND ANALYSIS

This section explains in detail the test results and analyzes the performance of the network environment described in the last section. The test results are analyzed mainly from UDP and TCP. The data for the experiment are collected every one second. Moreover, the sampling for each index lasts for 360 s for each time. The experimental data of this work are all reliable data collected from several experiments.

A. UDP Traffic

1. Throughput

Fig.6 depicts the throughput comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling time and y-axis represents the throughput. As can be seen from the figure, the VMs supported by KVM, Xen-Project and XenServer show similar throughputs with the physical machines. Although KVM and XenServer occasionally have abnormal points, their throughput remain 32 Mbits/s overall. The throughput of Xen-Project perfectly coincides with that of the Linux (the physical machine).

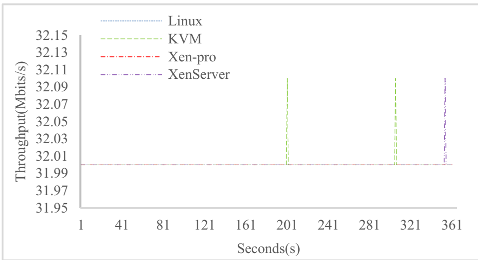

Fig.6. Throughput comparison of physical machines, KVM, Xen-project and Xenserver at 32M bandwidth in the UDP protocol test

However, the throughput of the network increases linearly with the increasing loads by modifying the bandwidth parameters of the Iperf. Fig.7 depicts the throughput comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling bandwidth and the y-axis represents the throughput. The linear relationship between the throughput and the load is broken down and starts to level off when the load increases to more than 500 M. The trend is seen clearly from Fig.7. The throughput trends of XenServer and Xen-Project are generally in agreement with that of the physical network, with the deviation under 1Mbits/s. While, the situation changes slightly when it comes to KVM. The drop of KVM is larger than that of other environments and the deviation from the physical machine reaches 30 M when the load is in the range from 500 M to 800 M. The reliability of the test results declines due to limitation of the testing tools if the load is more than 800 M. In general, the throughput of KVM is more significantly influenced by the load compared with the throughputs of Xen-Project and XenServer, showing a maximum deviation of 4%.

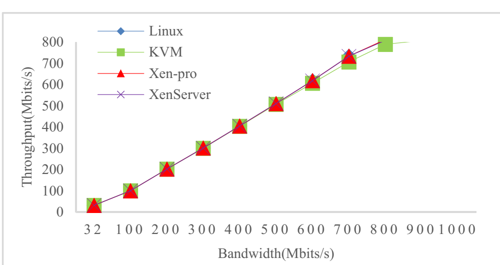

Fig.7. Throughput comparison of Physical machine, KVM, Xen-project and Xenserver at different bandwidths in the UDP protocol test

2. Jitter

Fig.8 depicts the jitter comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling time and the y-axis represents the jitter. It can be seen from that the jitter of XenServer is closer to the physical network than KVM and Xen-Project.

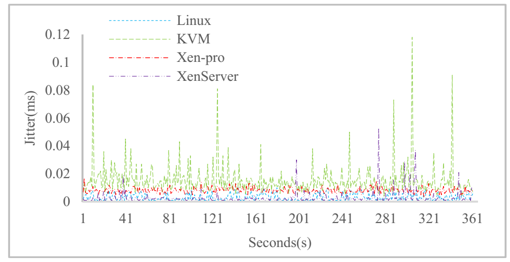

Fig.8. Jitter comparison of physical machines, KVM, Xen-project and Xenserver at 32M bandwidth in the UDP protocol test

Among them, KVM exhibits the largest jitter, which is 3.6 times that of the physical network. It is followed by Xen-Project, whose jitter is 1.5 times that of the physical network. XenServer is the best with the performance almost approximate to that of the physical machine. Therefore, in the case of a low-load, XenServer can be chosen to set up the network environment, which is likely to bring about the performance level of physical networks.

However, when the bandwidth of the Iperf and the load in the networks are changed, obvious variation can be observed among these network environments. Fig.9 depicts the jitter comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling bandwidth and the y-axis represents the jitter.

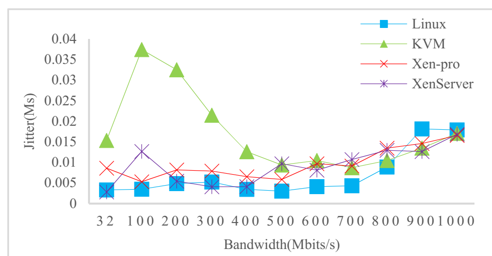

Fig.9. Jitter comparison of Physical machine, KVM, Xen-project and Xenserver at different bandwidths in the UDP protocol test.

The jitter of XenServer is very close to the physical network with deviation almost 0%; Xen-Project deviation is 160%; and the worst is KVM, whose deviation reaches 364%. When the bandwidth reaches 100M, XenServer jitter spikes and deviated 266% from the physical network, Xen-Project performance improves slightly whose deviation is reduced to 53%, the worst peak value of the KVM is appared that the deviation reaches 981 %. When the bandwidth is 100M ~ 400M, XenServer jitter reached the level of the physical network, Xen-Project continued to remain stable, while KVM jitter begin a sharp fall, and finally to the level under 100M. When the bandwidth is 600M ~ 800M, the jitter of XenServer, Xen-Project, KVM and the physical network starts to increase and the deviations relative physical network have a decreasing trend. By analyzing the jitter, the performance of Xen-Project is shown to be most steady and the trend is always in line with that of the physical network.

3. Packet Loss

Fig.10 depicts the packet loss comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling time and the y-axis represents the packet loss. It can be seen that the packet losses of the virtual networks separately supported by KVM, Xen-Project and XenServer are highly consistent with the physical network at 0%. The main reasons are low network loads and simple network structures.

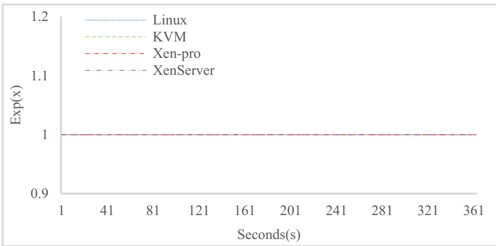

Fig.10. Packet Lost comparison of physical machines, KVM, Xen-project and Xenserver at 32M bandwidth in the UDP protocol test

With the virtual network bandwidth increasing, the virtual network build on KVM, Xen-Project separately begin to appear in packet loss. Packet loss frequency and quantity increases as the network bandwidth increasing. Fig.11 depicts the packet loss comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling bandwidth and the y-axis represents the packet loss.

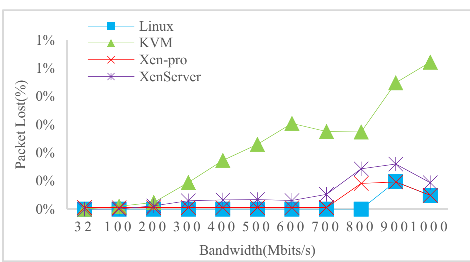

Fig.11. Packet Loss comparison of Physical machine, KVM, Xen-project and Xenserver at different bandwidths in the UDP protocol test

Within 200M, KVM, Xen-Project and XenServer have the same packet loss rate with physical network at 0%. Between 200M ~ 600M, KVM bandwidth, packet loss rate starts to increase linearly with each 100M increase 0.07%; Xen-Project dropout rate increased slightly, but the deviation always keeps within 0.0055%; XenServer packet loss rate is also slightly rises, the deviation remains within 0.03%. When the bandwidth exceed 700M, Xen-Project and XenServer packet loss rate increases rapidly, but still far less than the KVM packet loss rate. Only the bandwidth is high than 800M physical network packet loss rate began to increase obviously.

4. RTT

Fig.12 depicts the RTT comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling time and the y-axis represents the RTT.

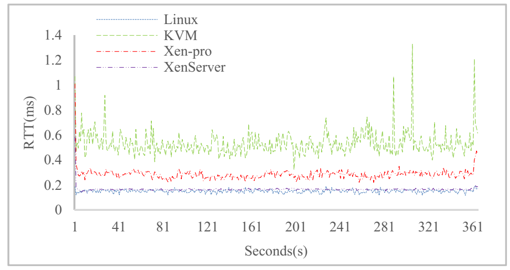

Fig.12. RTT comparison of physical machines, KVM, Xen-project and Xenserver at 32M bandwidth in the UDP protocol test

Packet round-trip delay between the physical network and virtual machines supported by XenServer are very close. On average, KVM delay is almost 350% of the physical network; virtual network supported by Xen-Project is 187% of physical network; while the RTT deviation of XenServer from the physical network is about 0%. Overall, KVM packet delay is the largest and the most unstable. Although Xen-Project is better than KVM, but much worse than that in the physical network and XenServer.

Fig.13 depicts the RTT comparison of physical machines, KVM, Xen-Project and XenServer at 32M in UDP protocol test, whose x-axis represents the sampling bandwidth and the y-axis represents the RTT. By modifying the bandwidth parameters of the Iperf and changing the network loads, the RTTs of UDP messages for the VMs supported by KVM, Xen-Project and XenServer are compared with that of the physical networks. It can be clearly seen that the RTT of KVM is maximum within 600M of the bandwidth and declines slightly with the increasing loads but on the whole, it is still deviated 221% from the physical network.

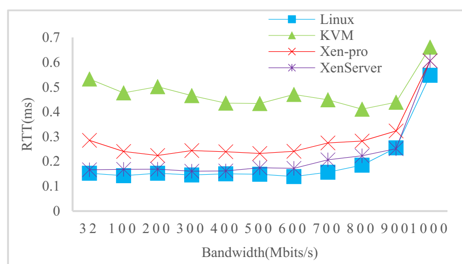

Fig.13. RTT comparison of Physical machine, KVM, Xen-project and Xenserver at different bandwidths in the UDP protocol test

The time delays of Xen-Project, XenServer and the physical machine network increase slightly with the increasing network loads, while generally remain stable, with deviation 65.4%, 13.8% separately. After more than 600M, all network delay are increasing rapidly, KVM, Xen-Project and XenServer three relative physical network deviations are 71.2%, 29.9% and 12.4%. From the data packet delay can be seen that XenServer is closer to the physical network and better network performance compared with KVM, Xen-Project.

B. TCP Traffic

1. Throughput

Fig.14 depicts the throughput comparison of physical machines, KVM, Xen-Project and XenServer in the TCP protocol test, whose x-axis represents the sampling time and the y-axis represents the throughput. It can be seen that the throughput of the physical network constructed based on the physical machine is very close to that of the virtual networks supported by KVM and Xen-Project. Among the three kinds of VM networks, Xen-Project is closest to the physical network and their standard deviations are 0.

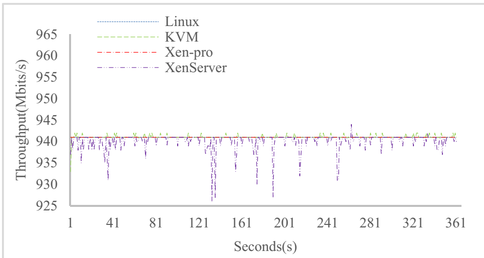

Fig.14. Throughput comparison of physical machines, KVM, Xen-project and Xenserver in the TCP protocol test

Although there is little difference in the performance between KVM and the physical network, it can be seen from the figure that the fluctuation of the KVM is larger than that of Xen-Project, and the standard deviation of KVM is 0.528. The throughput of XenServer fluctuates most obviously and its standard deviation is 2.045. Therefore, the throughput of Xen-Project is the best and most stable during the TCP communication.

2. RTT

Fig.15 depicts the RTT comparison of physical machines, KVM, Xen-Project and XenServer in the TCP protocol test, whose x-axis represents the sampling time and the y-axis represents the RTT. The time delays of the VMs separately supported by KVM, Xen-Project and XenServer differ greatly.

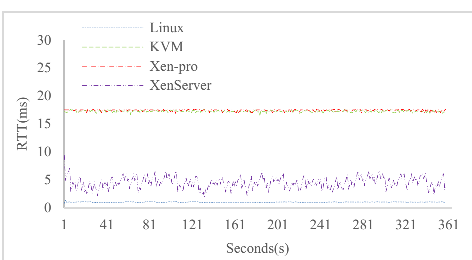

Fig.15. RTT comparison of physical machines, KVM, Xen-project and Xenserver in the TCP protocol test

Among them, KVM and Xen-Project present similar RTTs, at 17.2ms and 17.3ms, respectively. In addition, the deviation between them is about 0.1ms and their standard deviations are 0.86 and 0.88, respectively. Although the RTT of XenServer is small, there is a great difference compared with that of the physical network. Moreover, the RTT of XenServer fluctuates obviously. The average value and the standard deviation of the time delay of XenServer are 4.5ms and 1.14, respectively, while those of the physical network are 0.99ms and 0.05, respectively.

VI. CONCLUSION

The work analyzes the importance of virtual networks in the current and future applications. In addition, it also describes the architecture and data communication process concerning three common VMMs, including KVM, Xen-Project and XenServer. In order to evaluate the deviation of these three VMMs in the performance of virtual networks with the physical network, simple testing scenarios are designed for the virtual networks. In addition, the performances of the virtual networks supported by different VMMs are compared. Although the implementation principle and method of them are basically identical from the perspective of the architecture, the test results present some differences:

1. KVM, Xen-Project and XenServer perform very well in low-load networks and show little difference with that of the physical network.
2. The test result of UDP traffics shows that the performance of the virtual network built based on KVM is worse than that based on Xen-Project and XenServer.
3. KVM, Xen-Project and XenServer are less tolerant to high-load scenarios, where their performances reduce easily.
4. The VMMs have little support to TCP, while the throughput and the RTT of the virtual networks differ greatly with those of the physical machine, so they need to be optimized.
5. Compared with XenServer, the performance of Xen-Project is also affected by other factors apart from Xen-hypervisor, including the dom-0 regulated operating system and the integration methods for management tools.

The performance of virtual networks built for physical equipments is inevitably lower than that of the physical network. In order to improve the performance of virtual networks, apart from optimizing the VMMs and selecting better scheduling algorithms, other approaches can also be employed. For example, the optimization of the path of the data forwarding and the modification of the installation and integration method of the VMMs can be considered as well. In addition to these, the forwarding in the virtual network can also be managed to improve the performance of a network by introducing the controller on the control layer of SDN. This work provides a new perspective for optimizing and improving the performance of virtual networks in the future.

ACKNOWLEDGMENT

This work is partially supported by the National Program on Key Basic Research Project of China (973 Program) under Grant No. 2012CB315800, the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant No. 2012BAH01B00, the Strategic Priority Research Program of the Chinese Academy of Sciences under grant No. XDA06010306, the National Natural Science Foundation of China under Grant No. 61303241.

REFERENCES

[1]. Virtualization. https://en.wikipedia.org/wiki/Virtualization

[2]. N. Omnes, M. Bouillon, G. Fromentoux, & O. Le Grand, “A programmable and virtualized network & it infrastructure for the internet of things: How can nfv & sdn help for facing the upcoming challenges,” International Conference on Intelligence in Next Generation Networks, 2015, pp. 64-69.

[3]. MobilityFirst Future Internet Architecture Project Overview. http://mobilityfirst.winlab.rutgers.edu/

[4]. NAMED DATA NETWORKING. http://named-data.net/

[5]. eXpressive Internet Architecture. https://www.cs.cmu.edu/~xia/

[6]. GENI: Global Environment for Network Innovations. http://www.geni.net.

[7]. FIRE: Future Internet Research and Experimention. https://www.ict-fire.eu/

[8]. Sea Cloud Innovation Enviroment. http://scie.ac.cn/view/scie.jsp;jsessionid=726846FB71C250DB127EB33B311A3B56

[9]. R. Shea, F. Wang, H. Wang, & J. Liu, “A deep investigation into network performance in virtual machine based cloud environments,” Proceedings - IEEE INFOCOM, 2014, pp.1285-1293.

[10]. F. Gomez-Folgar, A. J. Garcia-Loureiro, T. F. Pena, J. I. Zablah, & N. Seoane, “Study of the KVM CPU performance of Open-Source cloud management platforms,” IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing IEEE, 2015, pp. 1225-1228.

[11]. B. R. Chang, H. F. Tsai, & C. M. Chen, “Evaluation of virtual machine performance and virtualized consolidation ratio in cloud computing system,” Journal of Information Hiding & Multimedia Signal Processing, vol. 3, April 2013, pp. 192-200.

[12]. Z. He, & G. Liang, “Research and evaluation of network virtualization in cloud computing environment,” 2012 Third International Conference on Networking and Distributed Computing. IEEE, 2012.

[13]. S. Khurana, & K. Marwah, “Performance evaluation of Virtual Machine (VM) scheduling policies in Cloud computing (spaceshared & timeshared),” Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013.

[14]. Libvirt. http://wiki.libvirt.org/page/Virtio

[15]. Ben-Yehuda, Muli, and Jon D. Mason. “The Xen hypervisor and its IO subsystem,” Presentation given at the 2005 IBM Systems and Storage Seminar, 2008.

[16]. Index of /jwl/TeachingActivities/Cloud/ref. http://140.136.149.173/jwl/TeachingActivities/Cloud/ref/Introduction%20to20%Xen.pdf

[17]. Li Jiaxiang. “Research on Optimization of Xen Network I/O Performance,” Huazhong University of Science and Technology, Wuhan, 2013.

[18]. O. Sefraoui, M. Aissaoui. & M. Eleuldj, “OpenStack: toward an open-source solution for cloud computing,” International Journal of Computer Applications 55, vol. 3, 2012.

[19]. CITRIX. https://www.citrix.com/

[20]. D. Bartholomew, "Qemu a multihost multitarget emulator,” Linux Journal, 2006.