

Analysis of Host Resources Utilization by OpenStack in Ubuntu Environment

Suryateja Pericherla ^{a*}

^a Department of CSE, Vishnu Institute of Technology, Andhra Pradesh, India

Abstract

Cloud computing is one of the frontier technologies, which over the last decade has gained a widespread commercial and educational user base. OpenStack is one of the popular open source cloud management platforms for establishing a private or public Infrastructure-as-a-Service (IAAS) cloud. Although OpenStack started with very few core modules, it now houses nearly 38 modules and is quite complex. Such a complex software bundle is bound to have an impact on the underlying hardware utilization of the host system. The objective is to monitor the usage of system resources by OpenStack on commodity hardware. This paper analyzes the effect of OpenStack on the host machine's hardware. An extensive empirical evaluation has been done on different types of hardware, at different virtualization levels and with different flavours of operating systems comparing the CPU utilization, memory consumption, disk I/O, network, and I/O requests. OpenStack was deployed using Devstack on a single node. The novel aspect of this work is monitoring the resource usage by OpenStack without creating virtual machines on commodity hardware. From the analysis of data, it is observed that standalone machine with Ubuntu server operating system is the least effected by OpenStack and thereby has more available resources for computation of user workloads.

Keywords:

Cloud Computing;
OpenStack;
Resource Utilization;
Devstack.

Article History:

Received:	05	September	2020
Accepted:	21	November	2020
Published:	01	December	2020

1- Introduction

Over the past decade cloud computing has become the de facto standard for dynamic provisioning of resources. Its other features like on demand access, ubiquitous nature, elasticity and pay-per-use model also lead to its success. Cloud computing provides services like Soft-ware-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), DataBase-as-a-Service (DBaaS) and more. Cloud computing can also be deployed in various ways like public cloud, private cloud, community cloud and hybrid cloud. Now-a-days companies are using services from clouds created by different vendors, making it a multi-cloud.

OpenStack is a cloud operating system for managing resources in a data center or an organization and for establishing a private or public cloud. It is the most popular choice now-a-days as seen in Figure 1, for providing Infrastructure-as-a-Service (IaaS). It was initially developed by NASA and Rackspace with few components. Now, many companies and developers are supporting the development of OpenStack. Although OpenStack contains approximately 38 components or modules, there are only few core components like nova, neutron, glance, keystone, swift and cinder. OpenStack also provides a dashboard through horizon module which allows administrator to easily monitor and provision resources to the clients.

There are five standard ways for deploying OpenStack: 1) using Juju Charms, 2) using Ansible in Docker containers, 3) using Ansible, 4) using TripleO and 5) using Devstack or Packstack. Both devstack and packstack are scripts which automatically deploys OpenStack on a single node. They are best suited for working with OpenStack for

* **CONTACT:** Suryateja.pericherla@gmail.com

DOI: <http://dx.doi.org/10.28991/esj-2020-01246>

© 2020 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

development purpose. Devstack works with debian versions of Linux like Ubuntu, and packstack works with Fedora and RedHat versions and their derivatives. Devstack was used for experimentation as Ubuntu operating system was selected. This paper is an extension of previously published work [1]. The research methodology followed is as shown in Figure 2.

The rest of the paper is organized as follows. Section 2 presents the related work involving OpenStack. Section 3 describes the experimental design, the hardware configuration and operating system choice. Section 4 presents the results obtained from observation and discuss those results. Section 5 concludes the paper.

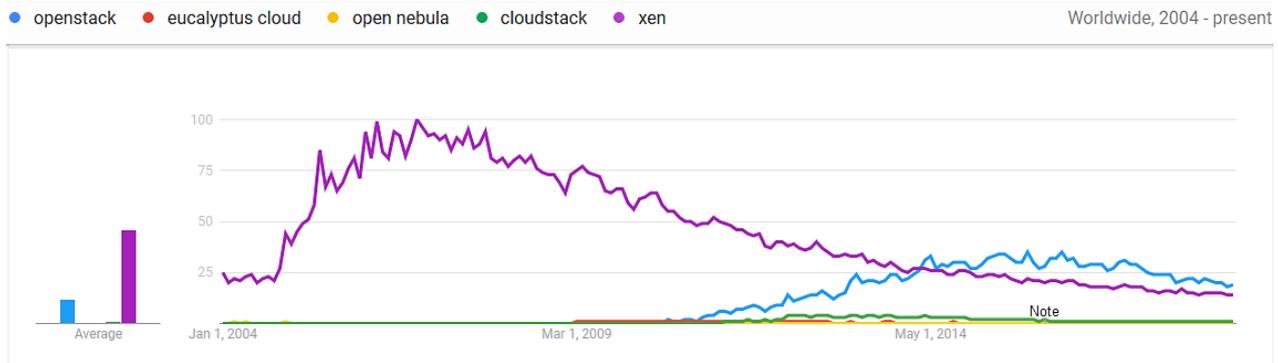


Figure 1. OpenStack, Eucalyptus, OpenNebula, CloudStack and Xen from Google Trends.

2- Related Work

Sefraoui et al. (2012) [2] provides a comparative study of cloud platforms like Eucalyptus, OpenNebula, and OpenStack. They aim to signify the importance of open source solutions over commercial solutions. They stress on the role of OpenStack in establishing private clouds. Nasim and Kassler (2014) [3] demonstrated the effect of underlying infrastructure on the performance of Open-Stack by conducting experiments on two separate test beds. They deployed OpenStack on virtual environment and also on dedicated hardware. From the experimental results, they conclude that performance of OpenStack on dedicated hardware is far greater than performance on virtual infrastructure. They measured the performance of CPU, data transfer rate and bandwidth in the OpenStack installations in the two test beds. Gebreyohannes (2014) [4] studied the performance of OpenStack Neutron, the net-work component in OpenStack. Using the IPERF bench-marking tool, the internal network performance of OpenStack was analyzed based on the parameters like packet loss, packet delay and throughput. From the experiments conducted, author concluded that OpenStack Neutron was scalable and it offers performance with no network bandwidth bottleneck.

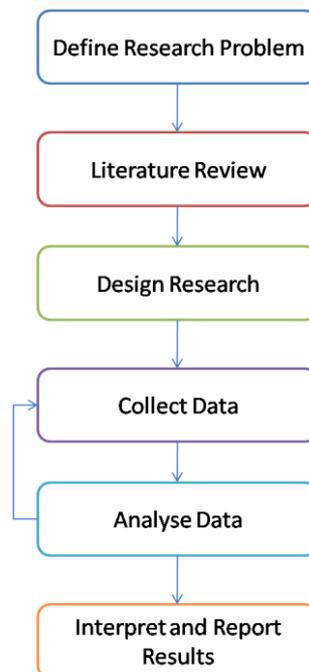


Figure 2. Research methodology.

Xu and Yuan (2014) [5] measured the performance of Quantum (later renamed to Neutron) which was the network component in OpenStack. They measured the performance by deploying Openstack using single-host plan and on

multi-host plan and by designing three test cases. All the tests were conducted on the installations after the deployment of Hadoop. Authors concluded that the performance of multi-host plan has significantly improved when compared with single-host plan. Mohammed and Kiran (2014) [6] present an experimental report in which they provide guidelines for setting up a cloud computing environment. They used the commodity infrastructure in their university's lab to setup a cloud computing environment. OpenStack was used as the cloud management platform. They used three nodes to setup the environment.

Grzonka et al. (2015) [7] focuses on the problem of utilizing the physical and virtual resource in OpenStack cloud platform effectively. Many test were conducted to measure the utilization of CPU and memory. Matrix multiplication was used for testing the performance of virtual resource. One important discovery in the experimentation was, increase in the number of virtual resource beyond a certain threshold, decreased the performance of the system. Raja and Rabinson (2016) [8] establish IaaS cloud using Openstack. Although the details of environment setup are not clear, they measured the utilization of CPU, memory and network. Tornyai et al. (2016) [9] conducted performance evaluation on a private cloud established using OpenStack. The aim of their study was to determine the internal behavior of OpenStack with respect to virtual machine's computing and networking capabilities. Authors used OpenStack component named Rally for automatic performance evaluation of OpenStack internals. Using different test cases they measured concurrency, stress and disk performance in the cloud.

Sha et al. (2015) [10] explain the effect of cloud deployment architecture on the cloud's performance by providing a high level formalization method called Performance Evaluation Process Algebra (PEPA), for modeling, analyzing and evaluating the cloud environment. The performance analysis of the cloud is carried out with respect to response time and the level of resource utilization. Callegati et al. (2016) [11] conducted several experiments on a cloud, managed using OpenStack, for evaluating the performance of Network Function Virtualization (NFV) and highlight its potentials and limitations. In the experiments they took care of both single tenant and multi-tenant scenarios. They also developed a visual tool which plots the different functional blocks created by OpenStack Neutron component. Through the experiments conducted, it was conclude that the Linux Bridge is a bottle-neck in the architecture, while Open vSwitch showed optimal behavior.

Shetty et al. (2017) [12] provided an empirical evaluation for measuring the performance of workloads over hypervisor based virtualization, container based virtualization and on bare metal machine. For hypervisor based virtualization OpenStack was selected, for container based virtualization Docker was used. Phoronix test suite was used for conducting different experiments for measuring the performance of CPU, RAM and disk in each of the aforementioned scenarios. The final conclusion was, bare machine provided maximum performance, followed by the container virtualization, followed by hypervisor based virtualization. Ahmad and Qazi (2018) [13] investigated the performance of CPU in OpenStack compute node and on a standalone system providing kernel-level virtualization. The experimentation was conducted with no VMs, one, two, three and four VMs. Later more experiments were conducted by varying the load and changing the operating systems and number of VMs. The operating systems that were tested are Ubuntu and CentOS. The experimental configurations were not provided which makes it unclear to understand the results of the experimentation done.

Table 1. Comparison of existing cases in literature.

References	Providers Compared	Features Compared
Sefraoui et al. (2012) [2]	Eucalyptus, OpenNebula, OpenStack	None
Nasim and Kassler (2014) [3]	OpenStack	CPU performance, Data transfer rate, Bandwidth
Gebreyohannes (2014) [4]	OpenStack	Network
Xu and Yuan (2014) [5]	OpenStack	Network
Mohammed and Kiran (2014) [6]	OpenStack	None
Grzonka et al. (2015) [7]	OpenStack	CPU, RAM
Raja and Rabinson (2016) [8]	OpenStack	CPU, RAM, Network
Tornyai et al. (2016) [9]	OpenStack	Concurrency, Stress, Disk
Sha et al. (2015) [10]	OpenStack	Response time, Utilization
Callegati et al. (2016) [11]	OpenStack	Networking
Shetty et al. (2017) [12]	OpenStack	CPU, RAM, Disk
Ahmad and Qazi (2018) [13]	OpenStack	CPU
Sajjad et al. (2018) [14]	OpenStack, Windows Azure	CPU, RAM, Disk, Network
Husain et al. (2020) [15]	OpenStack, Eucalyptus	Processing speed, Memory bandwidth, Disk I/O, Network
Mandal et al. (2019) [16]	OpenStack	Delay at controller node
Saghir and Massod (2019) [17]	OpenStack	Networking
Figliela et al. (2018) [18]	AWS, Azure, Google Cloud, IBM Cloud	Heterogeneous Cloud Functions
Noertjahyana et al. (2019) [19]	OpenStack	CPU

Sajjad et al. (2018) [14] conducted various experiments on two most popular cloud platforms namely OpenStack and Windows Azure. The features that were measured are CPU, memory, disk and network. For conducting experiments and evaluating the performance, several benchmark suites were used. Geekbench and LINPACK for CPU, RAMSpeed and STREAM for memory performance evaluation, IOzone for disk performance and finally for network performance, Iperf was used. The conclusion was, both OpenStack and Azure presented similar performance. But, OpenStack was considered as a better choice as it was free. Husain et al. (2020) [15] provided performance evaluation of OpenStack and Eucalyptus cloud platforms. The features measured were processor, memory, disk and network. For benchmarking, Linpack for processor, Stream for memory, Bonnie++ for disk I/O and IPerf for network were used. The results show that OpenStack gives better performance for disk I/O and Eucalyptus provides better performance for processor, memory and network operations. The summary of literature survey is provided in Table 1. There is no research available in the existing literature that had conducted detailed experiments for finding the effect of OpenStack on underlying commodity hardware without creating any virtual machines.

Mandal et al. (2019) [16] proposed an analytical model to measure the delay at controller node in the presence of one or more compute nodes. The deployed compute nodes use Advanced Message Queuing Protocol (AMQP) over Transmission Control Protocol/Internet Protocol (TCP/IP) to generate request messages which are processed by the controller node. Authors also derived the optimum service rate of controller to decrease the cost of the whole system. Saghir and Massod (2019) [17] proposed two technologies which overcomes the OpenStack's networking module Neutron's problem. Due to Neutron's centralized routing service, there is decrease in network throughput and bandwidth. The proposed technologies are OpenDayLight (ODL) and Distributed Virtual Routing (DVR). Authors evaluated the performance of ODL and DVR by creating virtual machines and benchmarking the network performance. Figiela et al. (2018) [18] evaluated the performance of cloud functions on major cloud providers like Amazon Web Services (AWS), Azure, Google Cloud, and IBM Cloud. Authors created their own benchmarking framework and also used another framework based on HyperFlow for measuring the performance of cloud functions. They had used CPU-intensive benchmarks like Mersenne, Twister and Linpack. Noertjahyana et al. (2019) [19] proposed creating a private cloud out of commodity hardware using OpenStack as the cloud hardware. The nodes used as part of the cloud were using Ubuntu Server 14.04 LTS operating system. The performance of the private cloud was evaluated using three benchmarks, namely: Sysbench, Linpack and Blender.

3- Experimental Design

OpenStack can be installed in several ways as discussed previously. All the experiments were conducted on OpenStack distribution named Queens which was installed using Devstack on a single node. So, all the OpenStack modules reside on a single machine (node). The modules installed are nova, neutron, glance, cinder, keystone, horizon and ceilometer. The experiments include two main scenarios.

3-1- Scenario 1 (S1)

Resource utilization is measured across a VM and a standalone machine with different underlying hardware. Resources monitored are CPU and memory. CPU utilization is monitored using *top* command (Linux) and memory usage is retrieved using *free* command (Linux). Hardware configuration of the machine containing the VM is different from the standalone machine. The host operating system of machine with VM is Windows 8.1 and the guest operating system is Ubuntu Server 16.04. VMWare Workstation was used for creating the VM as shown in Figure 3a. The operating system of standalone machine is Ubuntu Server 16.04 as shown in Figure 3b. Hardware configuration details of both VM and the standalone machine is given in Table 2.

Table 2. Hardware configuration.

	VM	Standalone
<i>Architecture</i>	x86_64	x86_64
<i># CPU(s)</i>	1	8
<i>Thread(s) per core</i>	1	2
<i>CPU Speed</i>	1.70GHz	3.40GHz
<i>Motherboard</i>	Intel Corporation	HP
<i>L1 Cache</i>	16 KB	256 KB
<i>L2 Cache</i>	-	1 MB
<i>L3 Cache</i>	-	8 MB
<i>RAM</i>	6 GB	8 GB
<i>Disk</i>	40 GB	931 GB
<i>Ethernet</i>	1 Gb/s	100 Mb/s



Figure 3. Scenario 1 configuration stack.

3-2- Scenario 2 (S2)

Resource utilization is measured across a VM and a standalone machine with same underlying hardware. Resources monitored are CPU, memory, disk I/O, network, and I/O requests. CPU utilization, disk I/O, network, and I/O requests are monitored using *dstat* command (Linux) and memory usage is retrieved using *free* command (Linux). The operating systems used are Ubuntu Server 16.04 as shown in Figure 4a. and Ubuntu Desktop 16.04 as shown in Figure 4b. The VM is also created on the standalone machine with the host operating system as Ubuntu Desktop 16.04 and guest operating system as Ubuntu Server 16.04 as shown in Figure 4c. The configuration of VM created in this scenario is similar to that of the VM created in scenario 1. VM was created using VirtualBox. Hardware configuration details are same as mentioned for standalone machine in scenario 1.

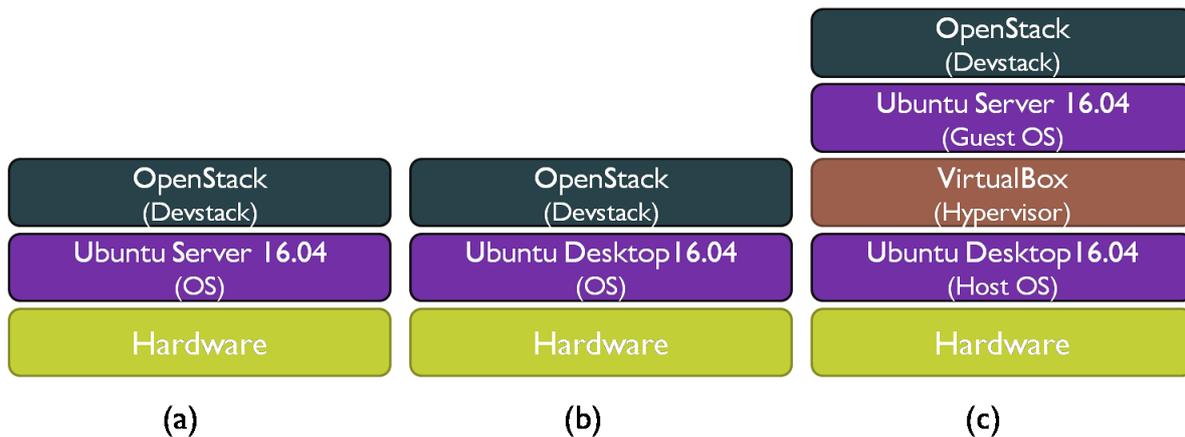


Figure 4. Scenario 2 configuration stack.

4- Results and Discussion

The results are described separately for the two scenarios mentioned in the previous section.

4-1- Scenario 1 Results

For scenario 1 (as mentioned in section 3), *top* command is used to monitor the CPU utilization. 2500 frames or instances are collected and plotted. Memory utilization is calculated using the *free* command and is given in Table 3.

Table 3. Memory utilization in scenario 1.

	VM		Standalone	
	Before	After	Before	After
Total (MB)	5949	5949	7892	7892
Free (MB)	5568	888	7543	2874
Utilization %	1.24	79.14	1.15	58.55

The CPU utilization in the VM before installing OpenStack is presented in Figure 5 and the CPU utilization in the VM after installing OpenStack is presented in Figure 6. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The CPU utilization in the standalone machine before installing OpenStack is presented in Figure 7 and the CPU utilization in the standalone machine after installing OpenStack is presented in Figure 8. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized.

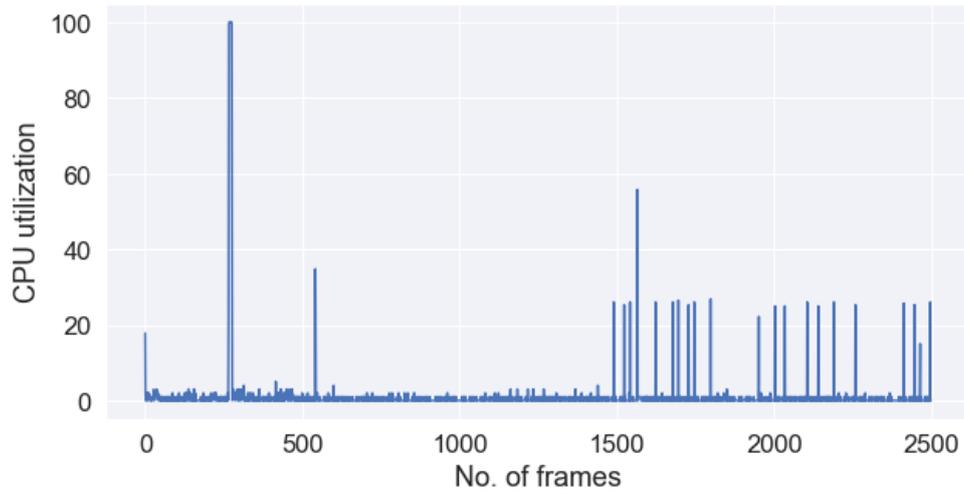


Figure 5. CPU utilization in VM before OpenStack installation.

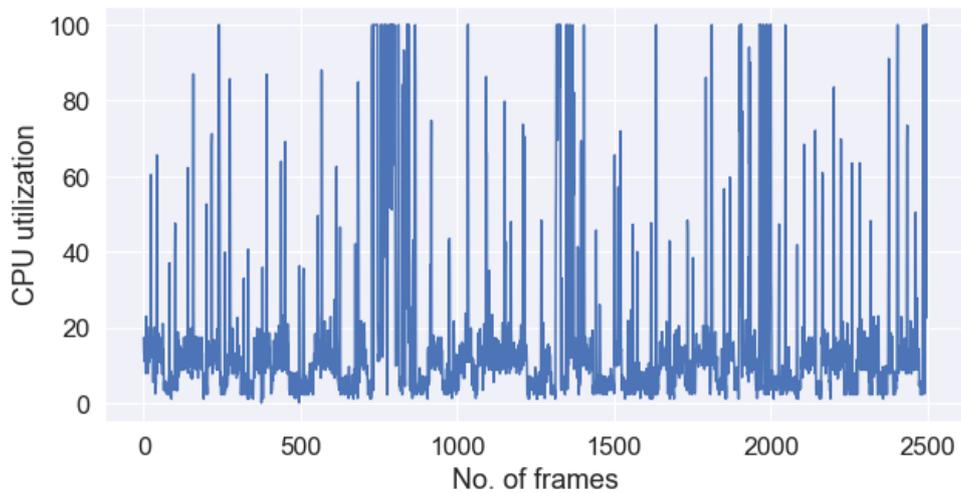


Figure 6. CPU utilization in VM after OpenStack installation.

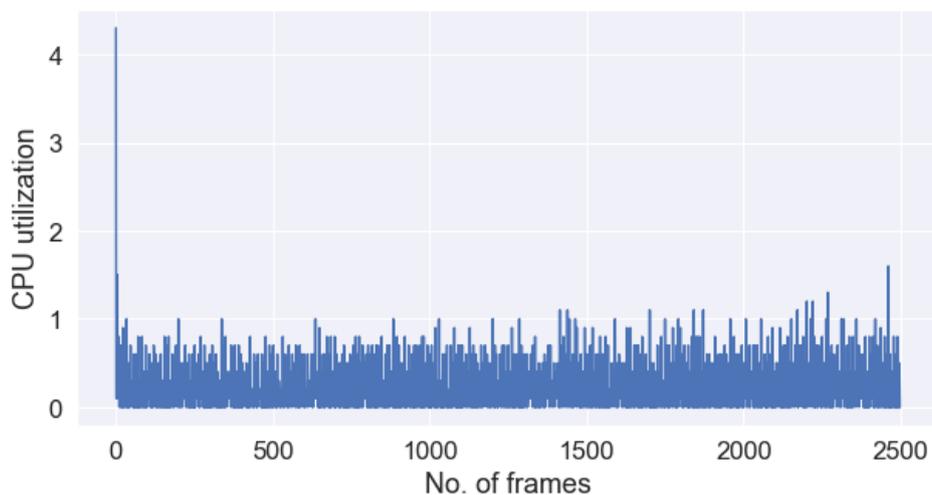


Figure 7. CPU utilization in standalone machine before OpenStack installation.

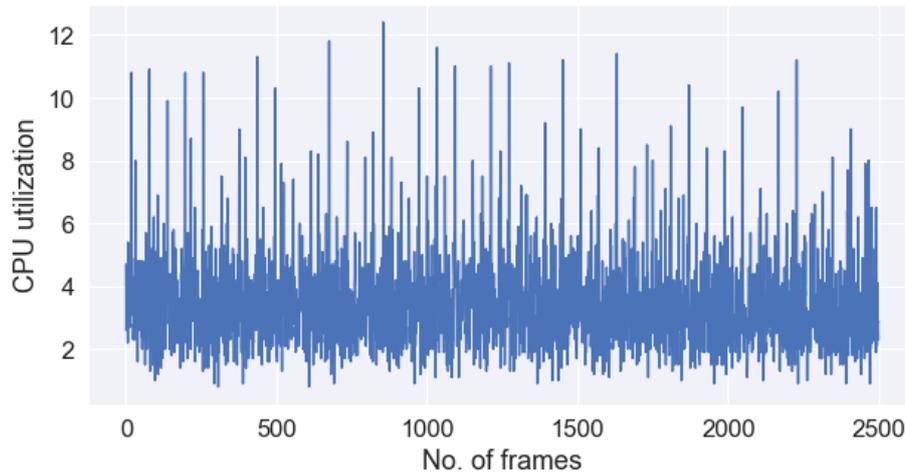


Figure 8. CPU utilization in standalone machine after OpenStack installation.

A combined plot of CPU utilization in VM before and after installing OpenStack is presented in Figure 9 and a combined plot of CPU utilization in standalone machine before and after OpenStack installation is presented in Figure 10. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. A combined plot of CPU utilization in VM and the standalone machine before installing OpenStack is given in Figure 11 and a combined plot of CPU utilization in VM and the standalone machine after installing OpenStack is given in Figure 12. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized.

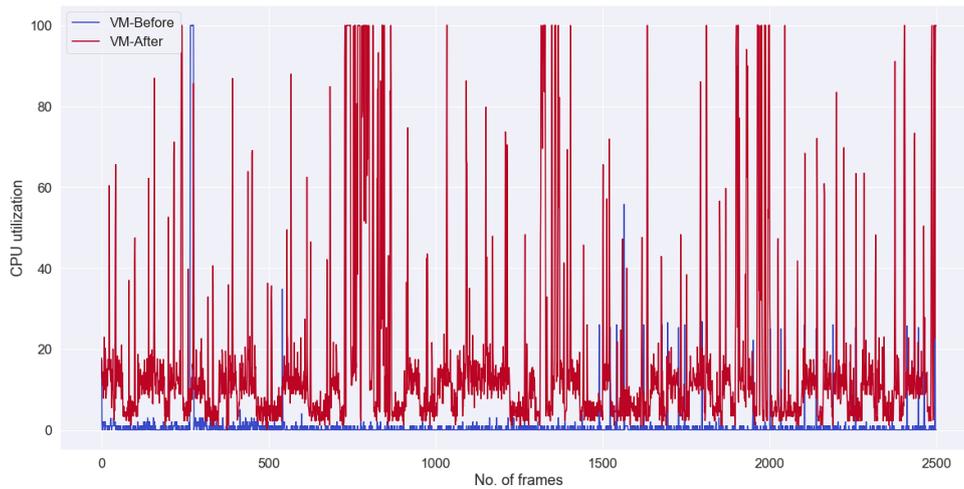


Figure 9. CPU utilization in VM before and after OpenStack installation.

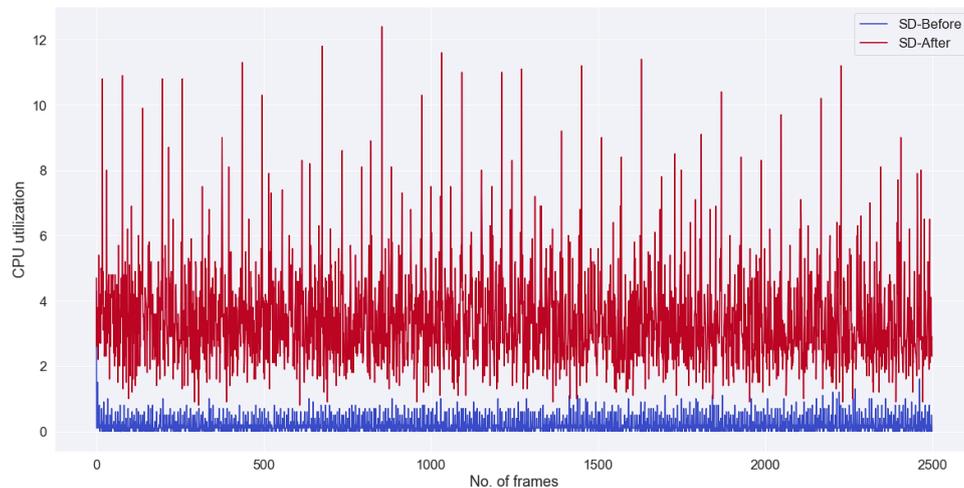


Figure 10. CPU utilization in standalone machine before and after OpenStack installation.

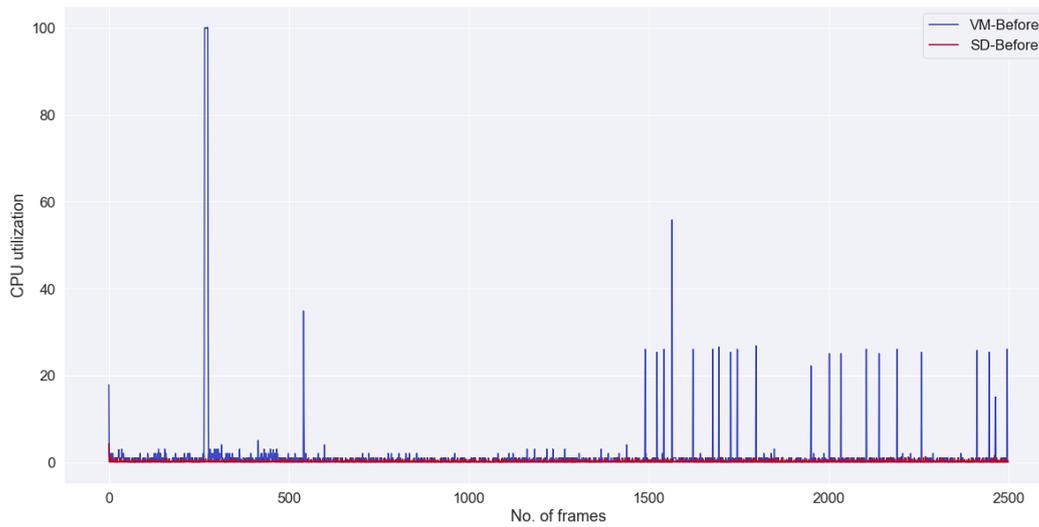


Figure 11. CPU utilization in VM and standalone machine before installing OpenStack.

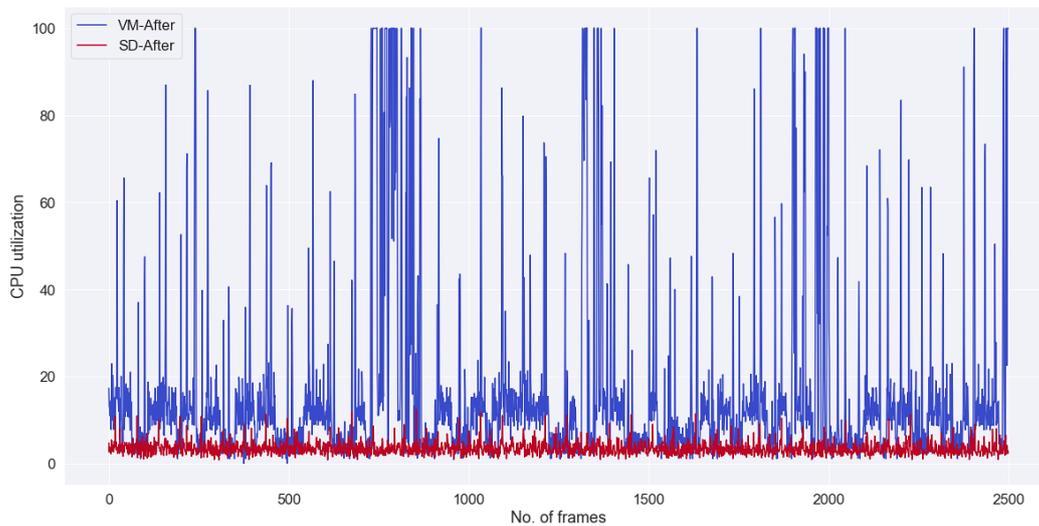


Figure 12. CPU utilization in VM and standalone machine after installing OpenStack.

4-2- Scenario 2 Results

For scenario 2 (as mentioned in section 3), *dstat* command is used to monitor the CPU utilization, disk I/O, network, and I/O requests. This command was executed approximately for one hour and 3600 instances or frames were collected. The underlying hardware for this entire scenario is same. Memory utilization is calculated using the *free* command and is given in Table 4.

Table 4. Memory utilization in scenario 2.

	VM		Standalone	
	Before	After	Before	After
<i>Memory</i>				
<i>Total (MB)</i>	5967	5967	7892	7892
<i>Free (MB)</i>	5596	1685	7543	2860
<i>Utilization %</i>	1.2	65.79	1.12	57.24

The CPU utilization in standalone machine with Ubuntu Server OS before installing OpenStack (S2.1) is presented in Figure 13. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes before installing OpenStack are given in Figure 14a and Figure 14b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent before installing OpenStack are presented in Figure 15a and Figure 15b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests before installing OpenStack are given in Figure 16a and Figure 16b.

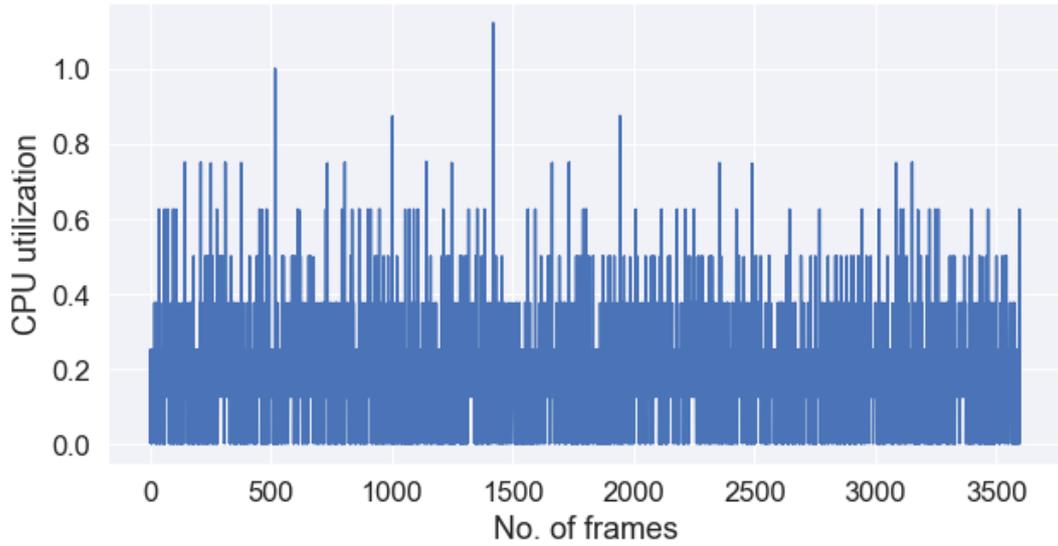


Figure 13. CPU utilization in S2.1.

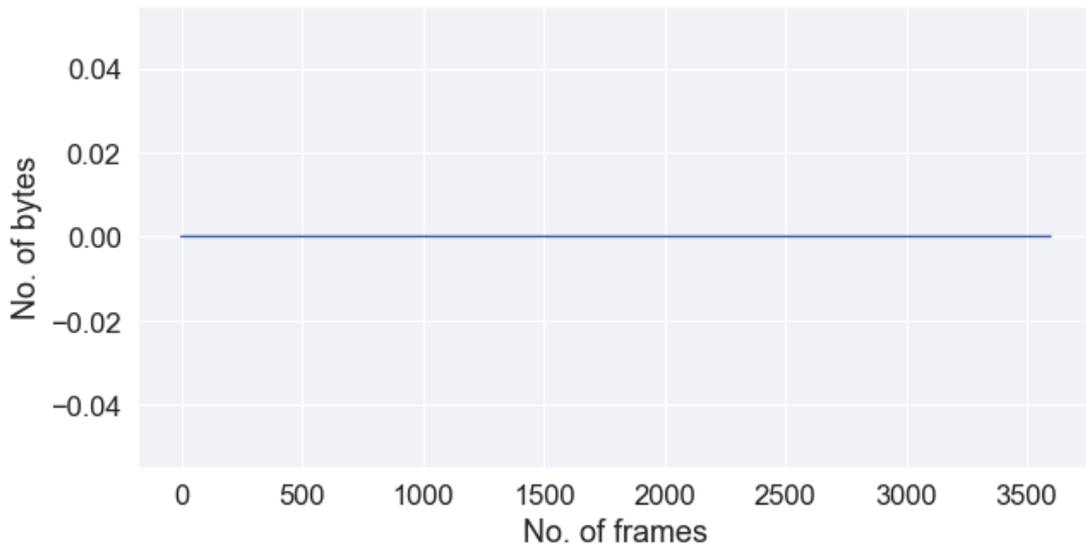


Figure 14a. Disk reads in S2.1.

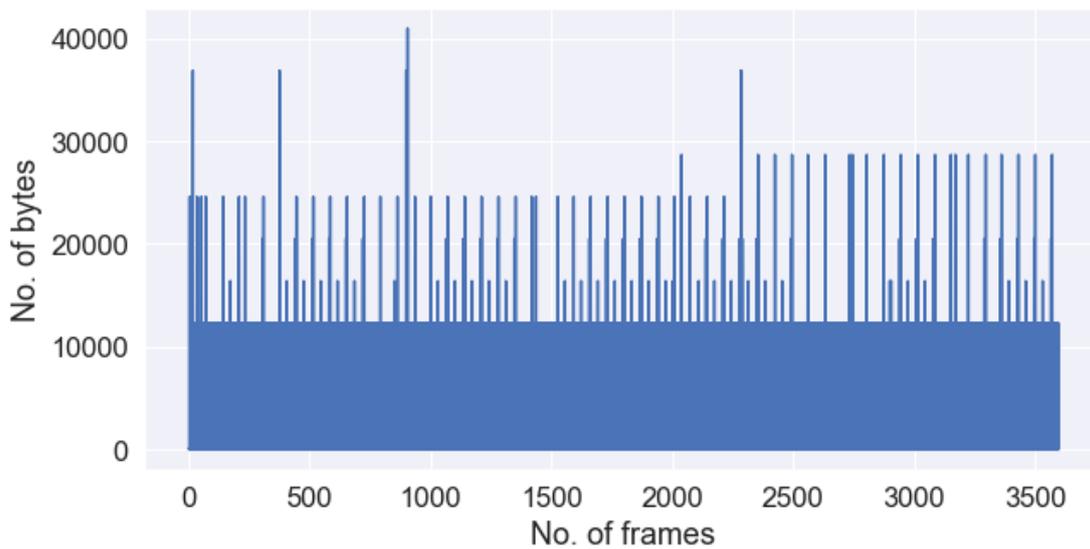


Figure 14b. Disk writes in S2.1.

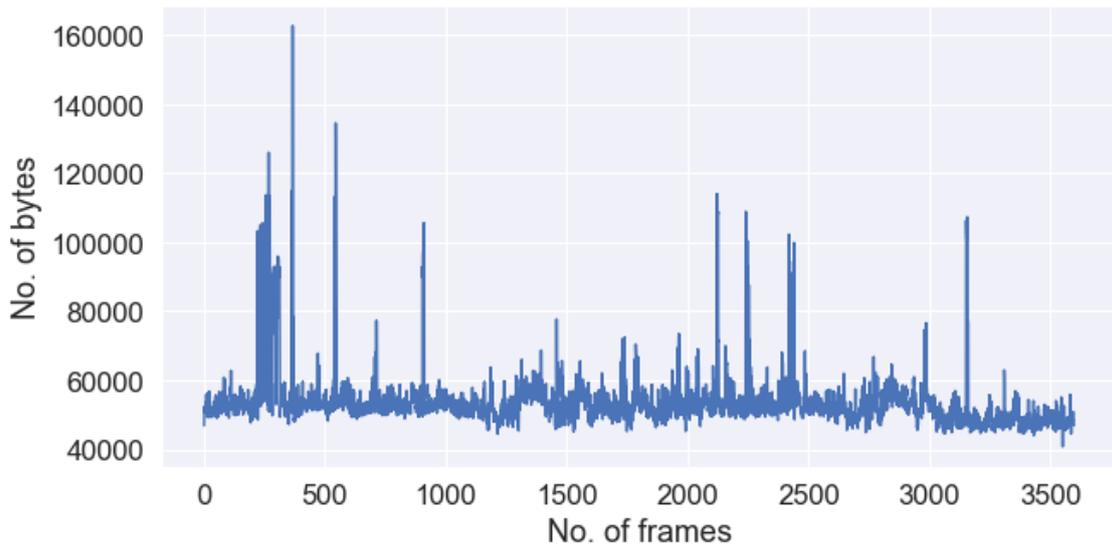


Figure 15a. Network bytes received in S2.1.

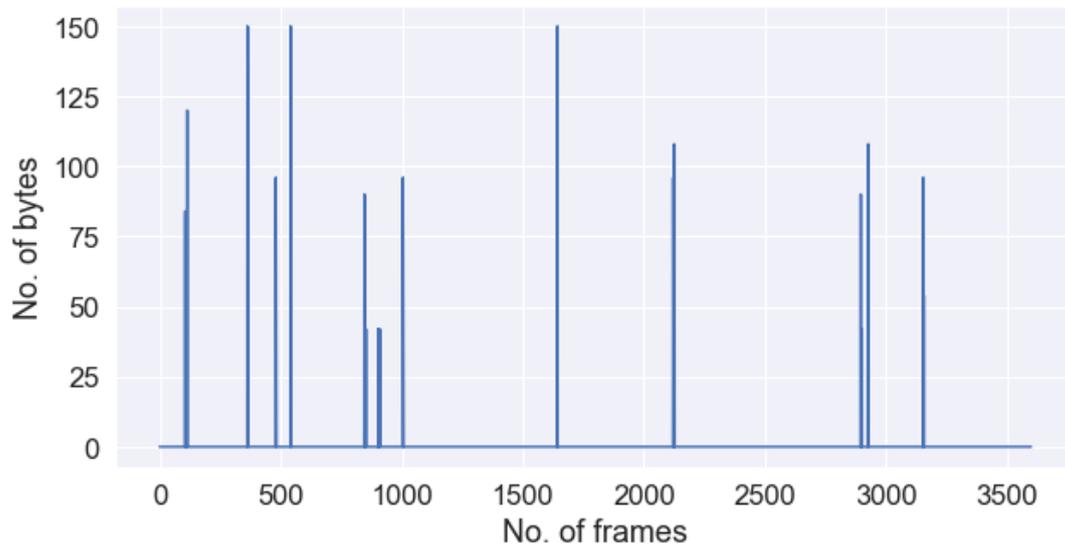


Figure 15b. Network bytes sent in S2.1.

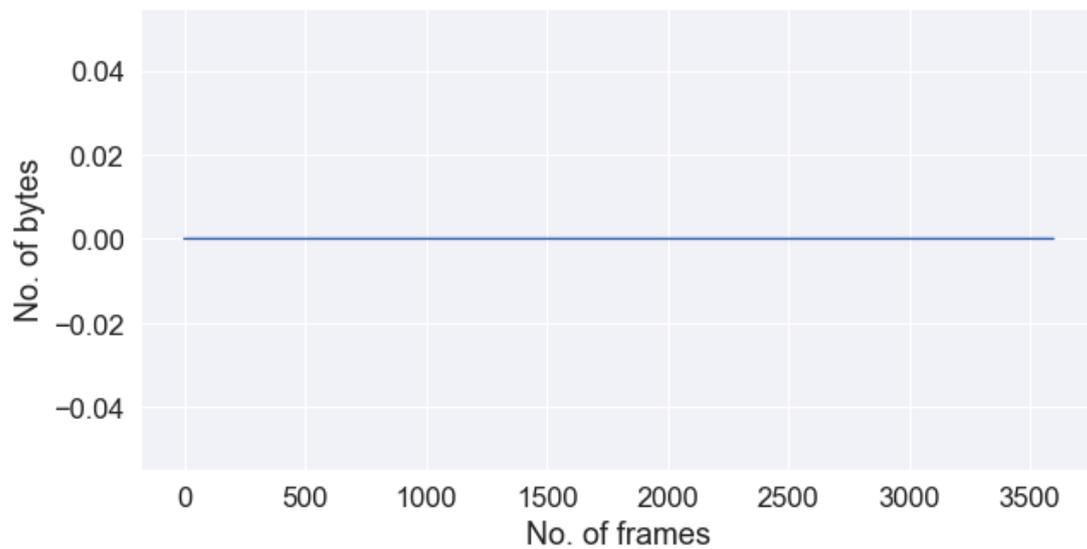


Figure 16a. I/O read requests S2.1.

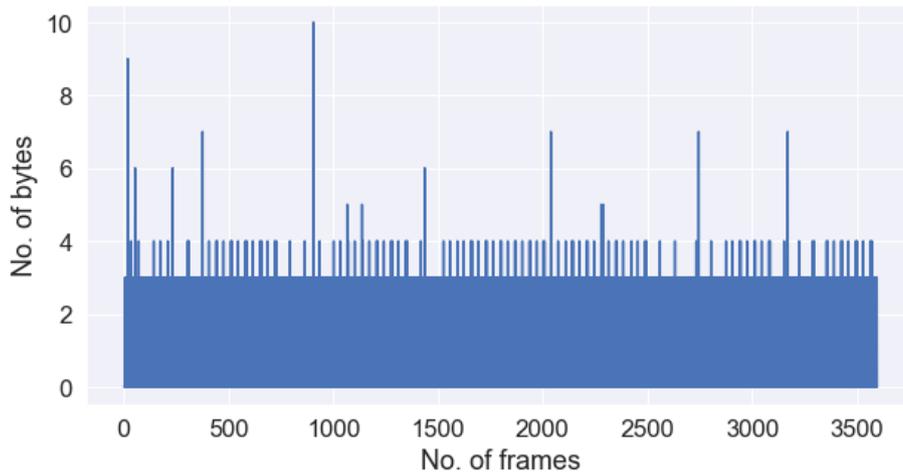


Figure 16b. I/O write requests S2.1.

The CPU utilization in standalone machine with Ubuntu Server OS after installing OpenStack (S2.2) is presented in Figure 17. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes after installing OpenStack are given in Figure 18a and Figure 18b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent after installing OpenStack are presented in Figure 19a and Figure 19b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests after installing OpenStack are given in Figure 20a and Figure 20b.

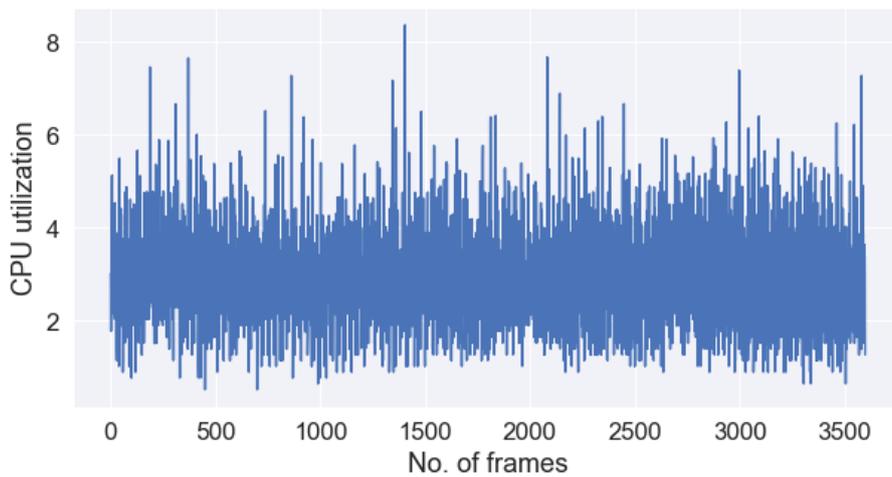


Figure 17. CPU utilization in S2.2.

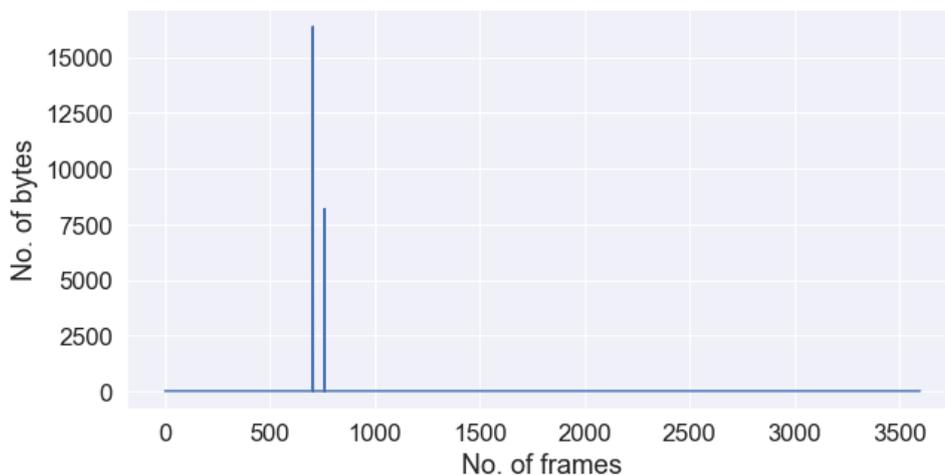


Figure 18a. Disk reads in S2.2.

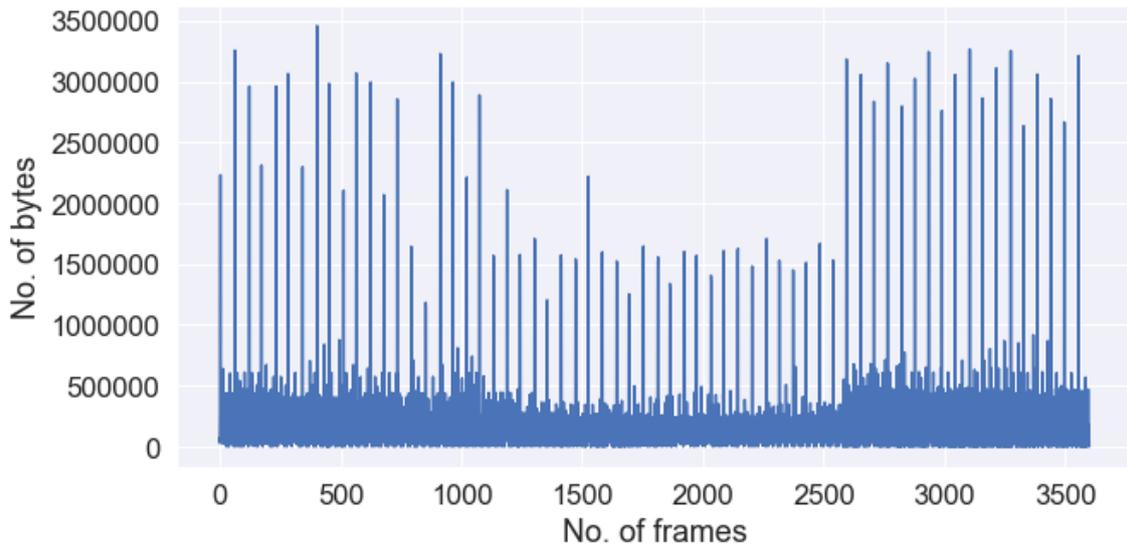


Figure 18b. Disk writes in S2.2.

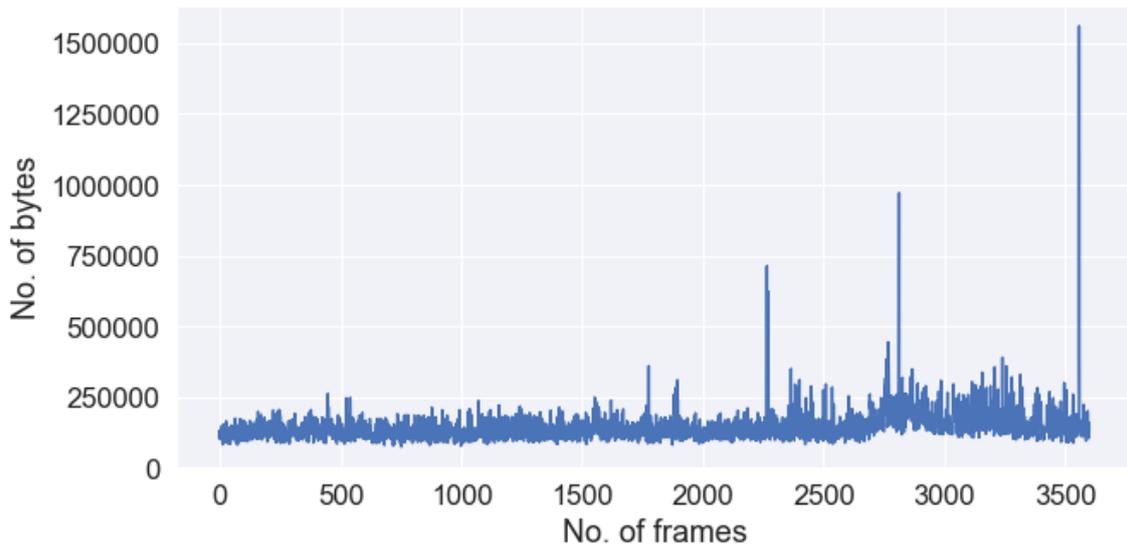


Figure 19a. Network bytes received in S2.2.

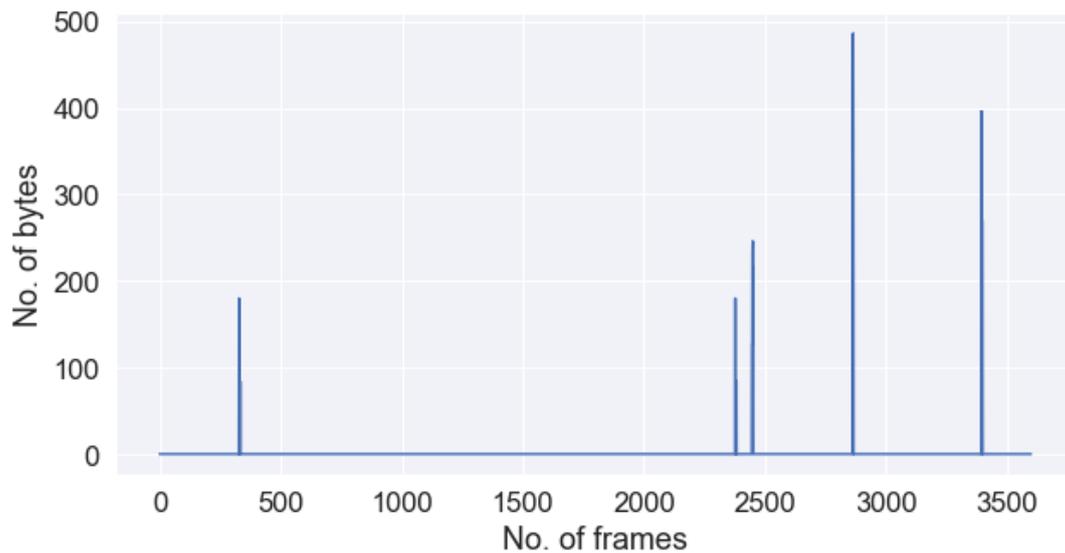


Figure 19b. Network bytes sent in S2.2.

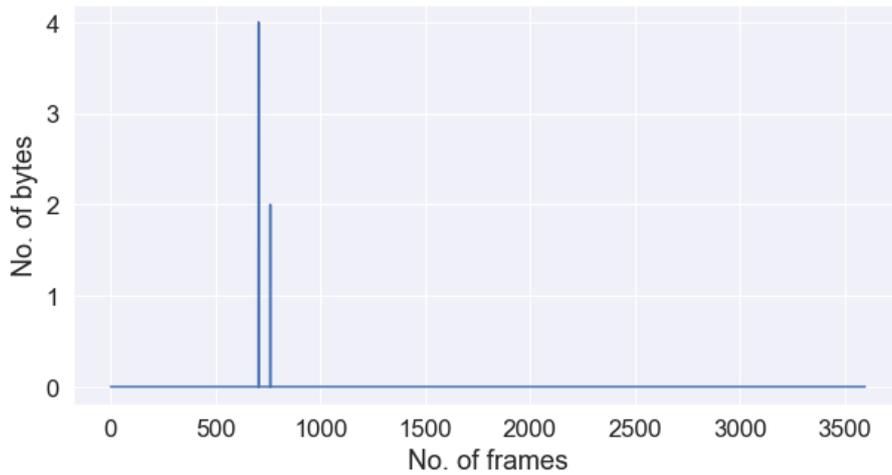


Figure 20a. I/O read requests S2.2.

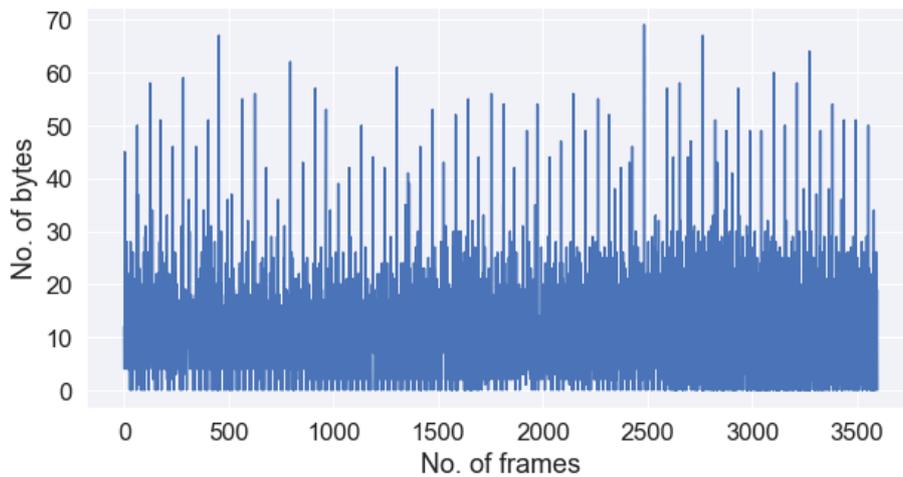


Figure 20b. I/O write requests S2.2.

The CPU utilization in standalone machine with Ubuntu Desktop OS before installing OpenStack (S2.3) is presented in Figure 21. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes before installing OpenStack are given in Figure 22a and Figure 22b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent before installing Open-Stack are presented in Figure 23a and Figure 23b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests before installing OpenStack are given in Figure 24a and Figure 24b.

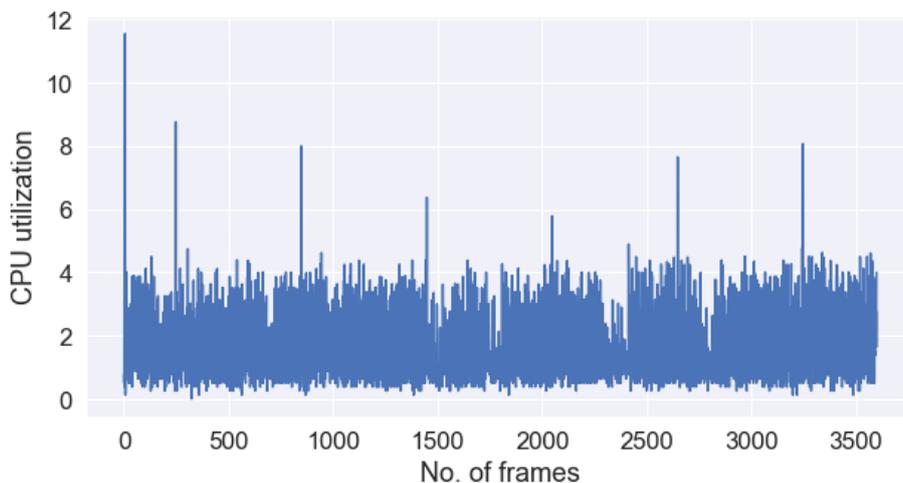


Figure 21. CPU utilization in S2.3.

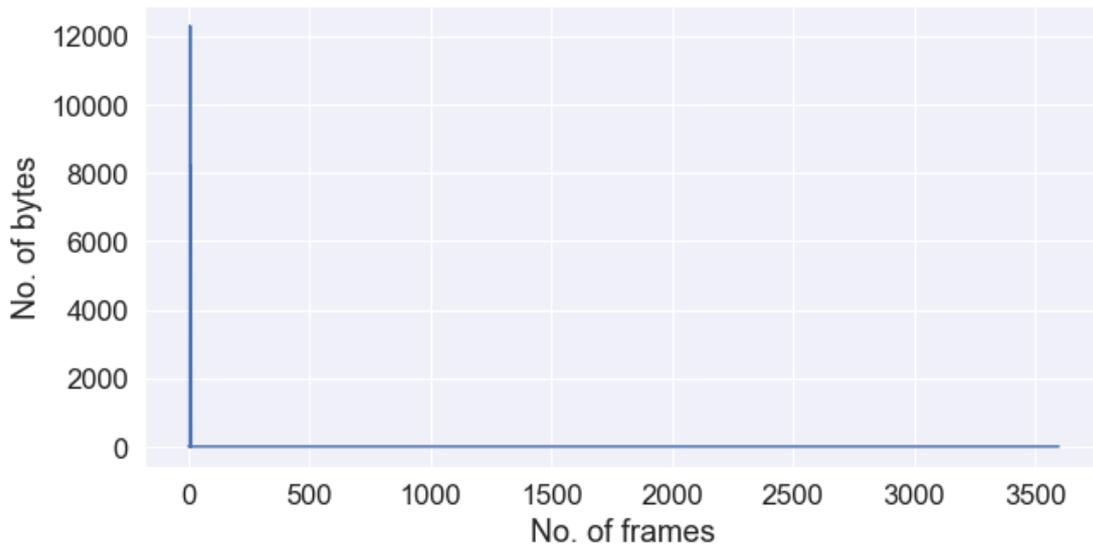


Figure 22a. Disk reads in S2.3.

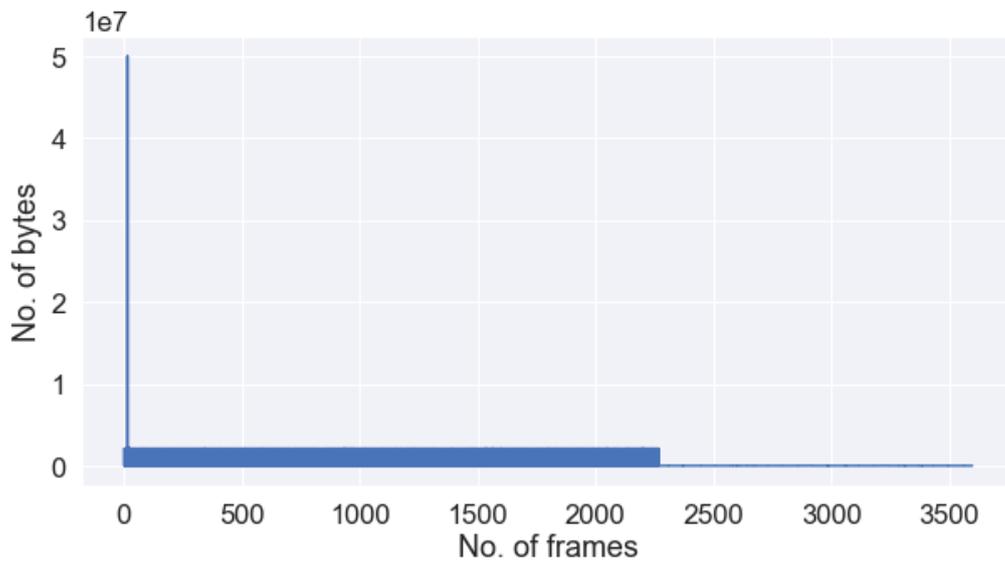


Figure 22b. Disk writes in S2.3.

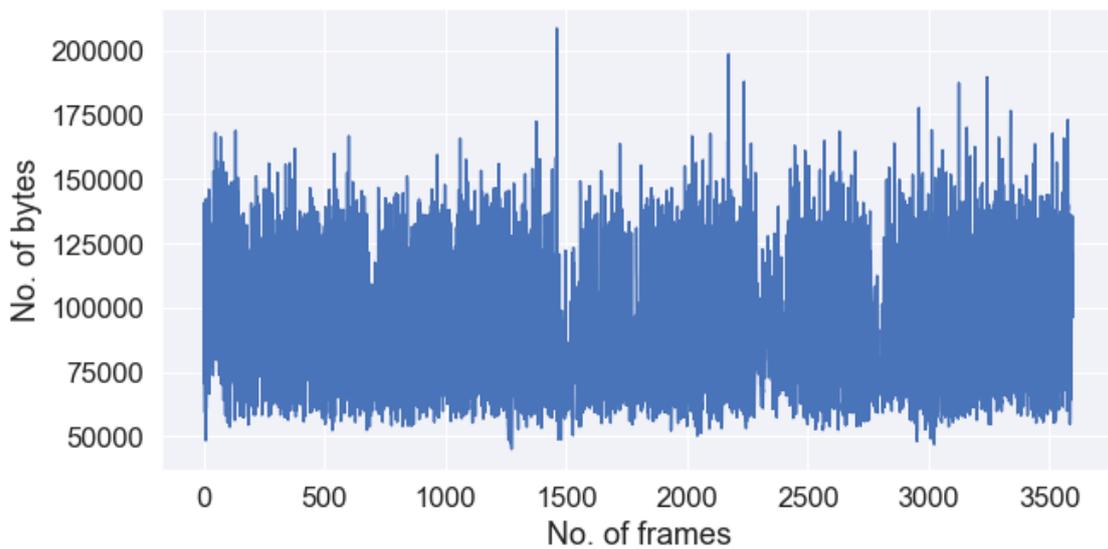


Figure 23a. Network bytes received in S2.3.

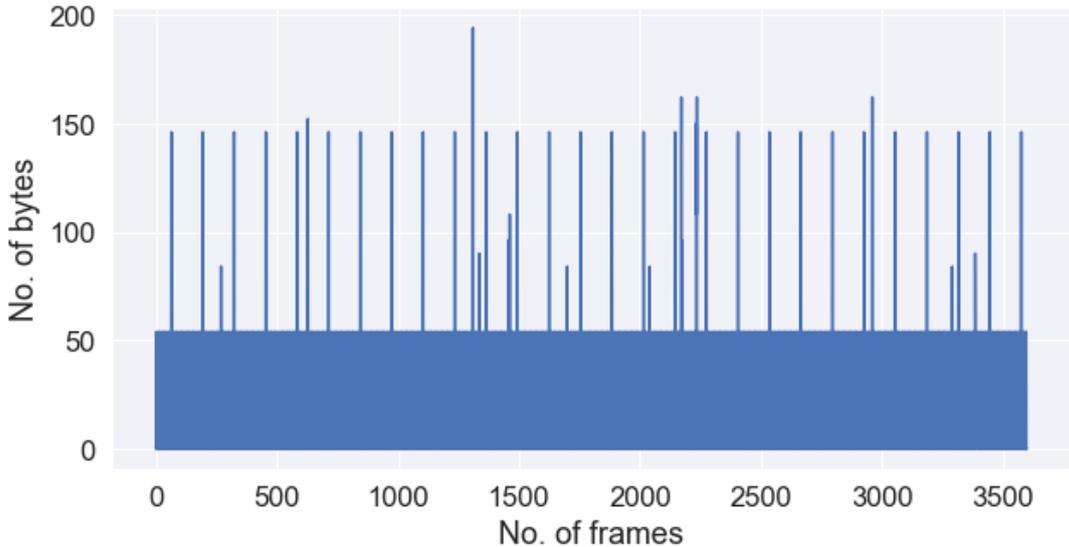


Figure 23b. Network bytes sent in S2.3.

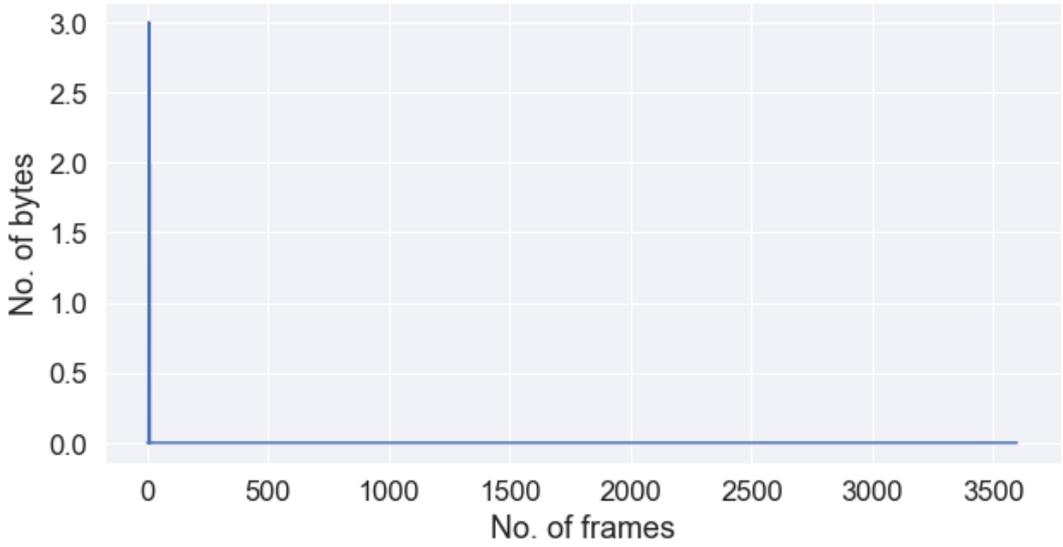


Figure 24a. I/O read requests S2.3.

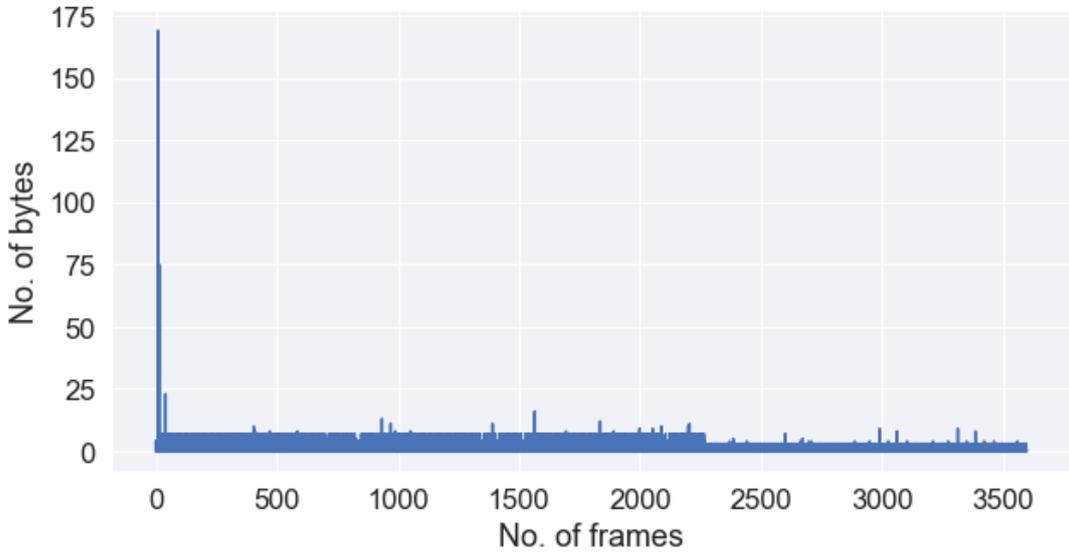


Figure 24b. I/O write requests S2.3.

The CPU utilization in standalone machine with Ubuntu Desktop OS after installing OpenStack (S2.4) is presented in Figure 25. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes after installing OpenStack are given in Figure 26a and Figure 26b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent after installing OpenStack are presented in Figure 27a and Figure 27b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests after installing OpenStack are given in Figure 28a and Figure 28b.

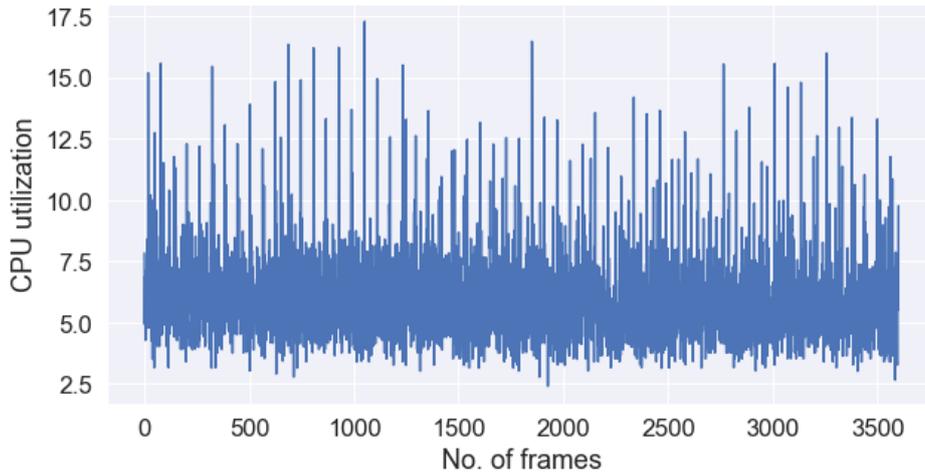


Figure 25. CPU utilization in S2.4.

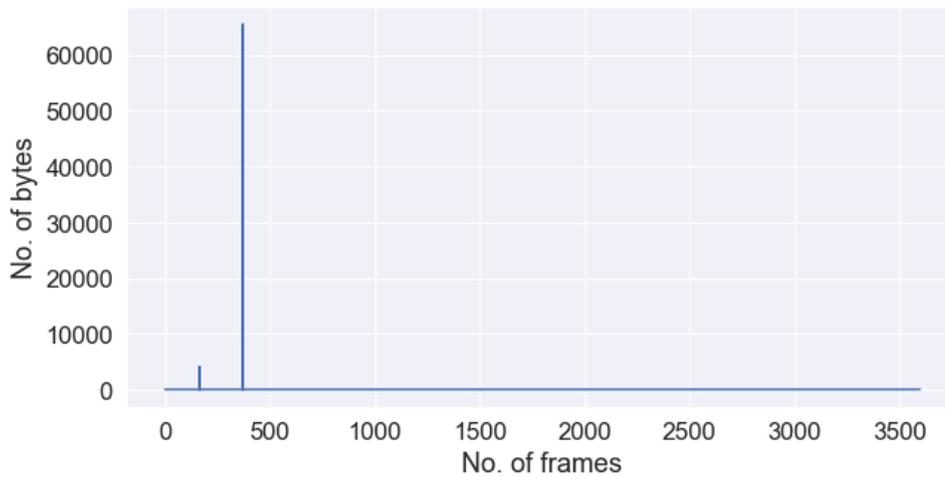


Figure 26a. Disk reads in S2.4.

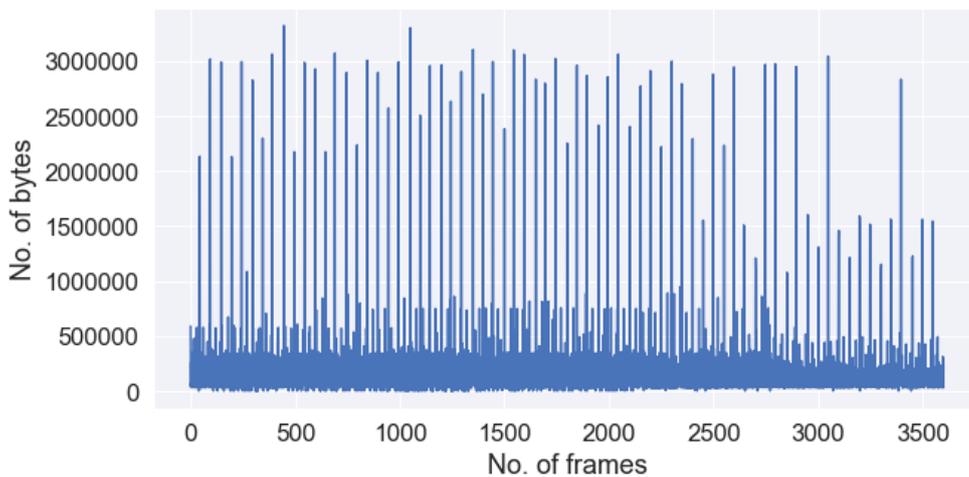


Figure 26b. Disk writes in S2.4.

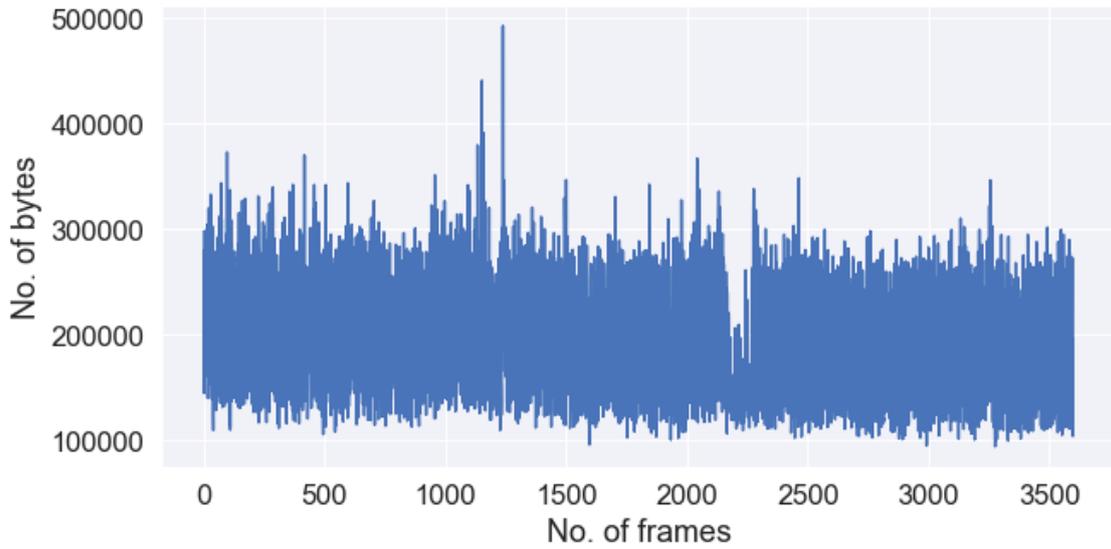


Figure 27a. Network bytes received in S2.4.

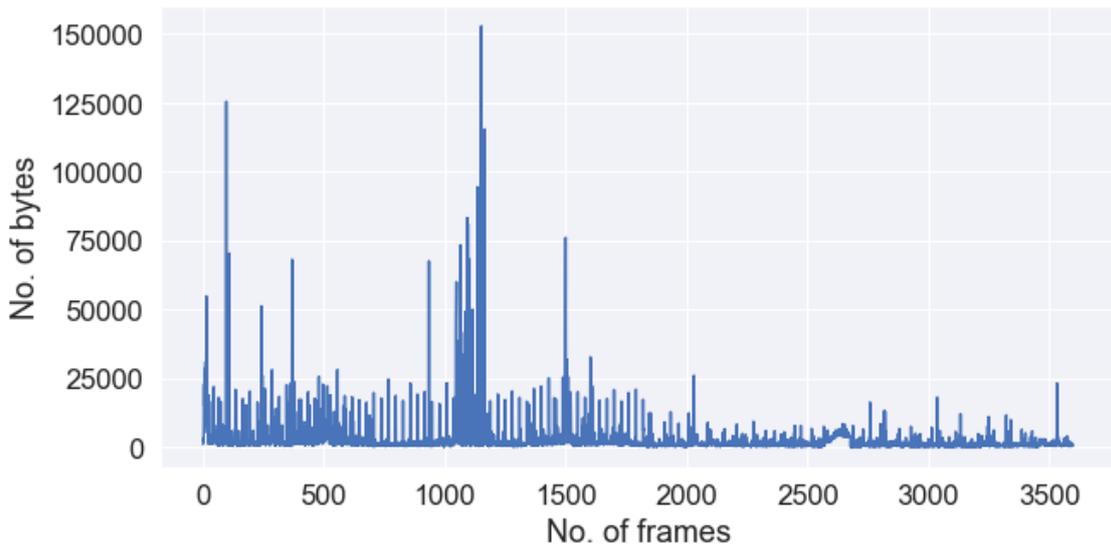


Figure 27b. Network bytes sent in S2.4.

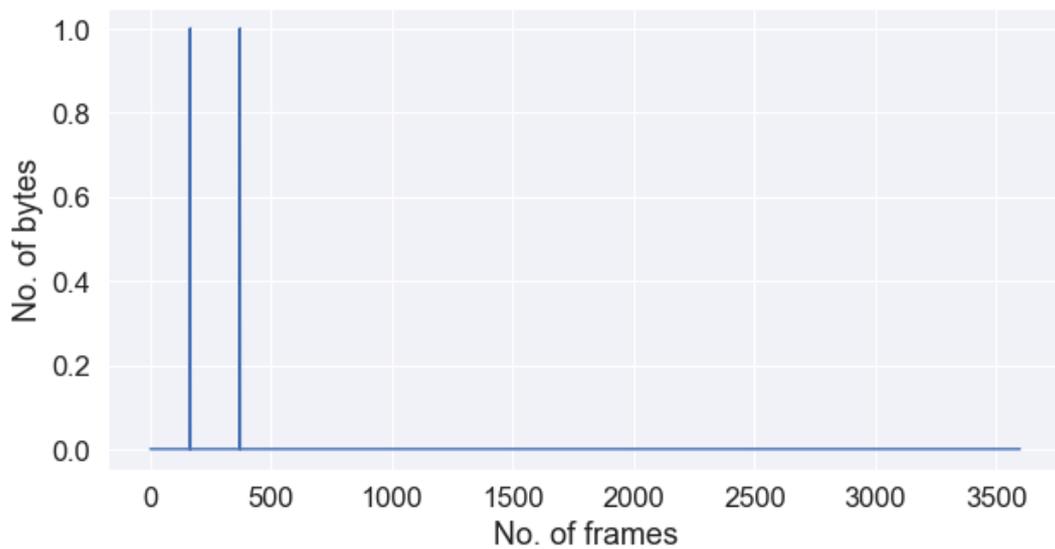


Figure 28a. I/O read requests S2.4.

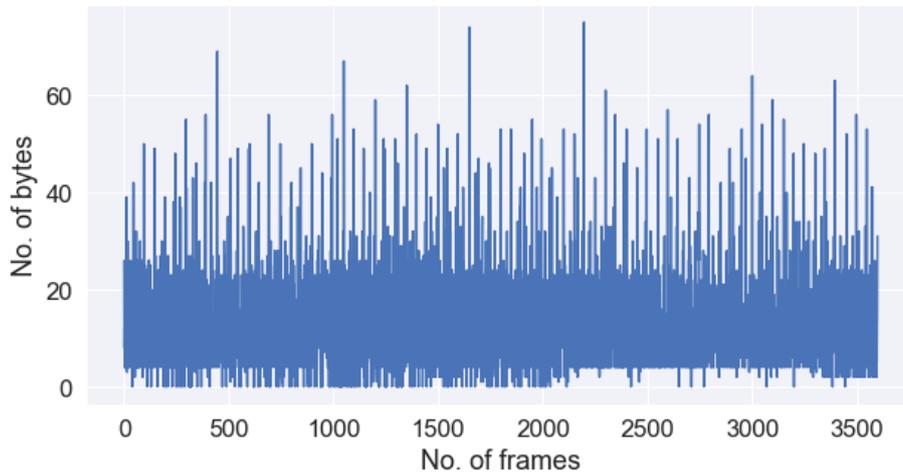


Figure 28b. I/O write requests S2.4.

The CPU utilization in VM with Ubuntu Desktop host operating system before installing OpenStack (S2.5) is presented in Figure 29. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes before installing OpenStack are given in Figures 30a and 30b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent before installing OpenStack are presented in Figures 31a and 31b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests before installing Open-Stack are given in Figures 32a and 32b.

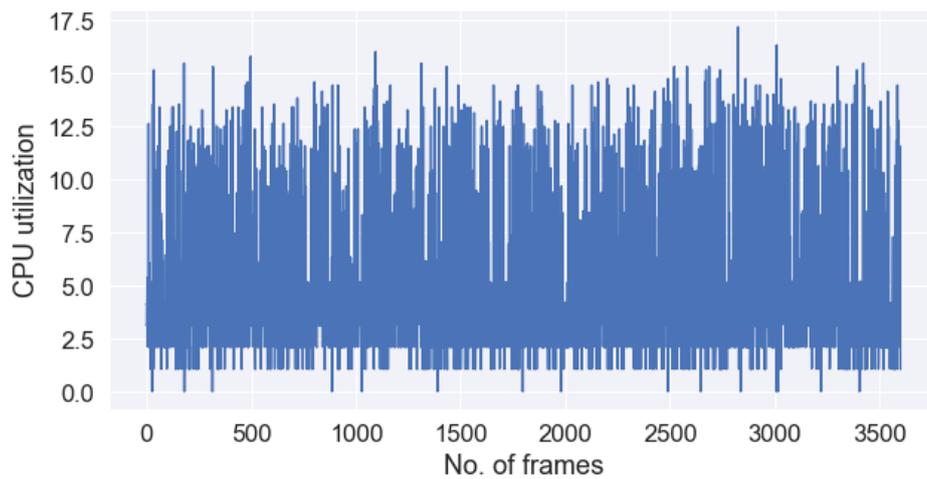


Figure 29. CPU utilization in S2.5.

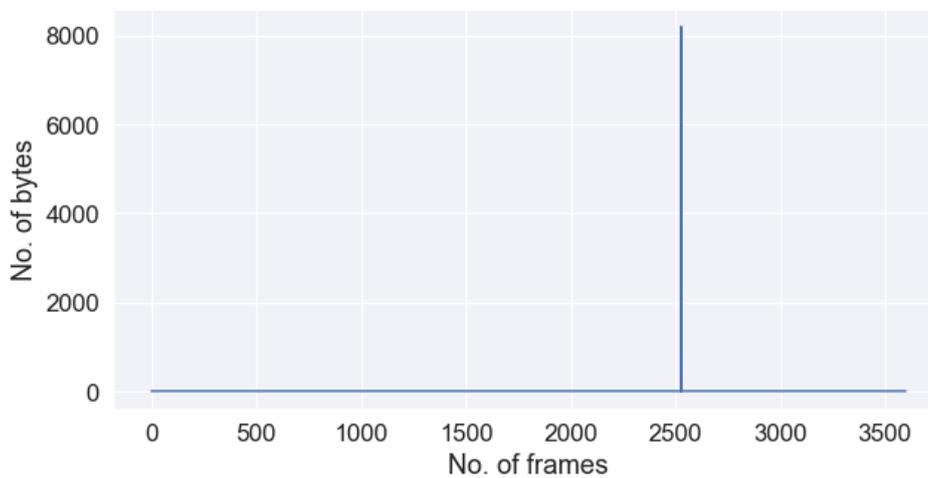


Figure 30a. Disk reads in S2.5.

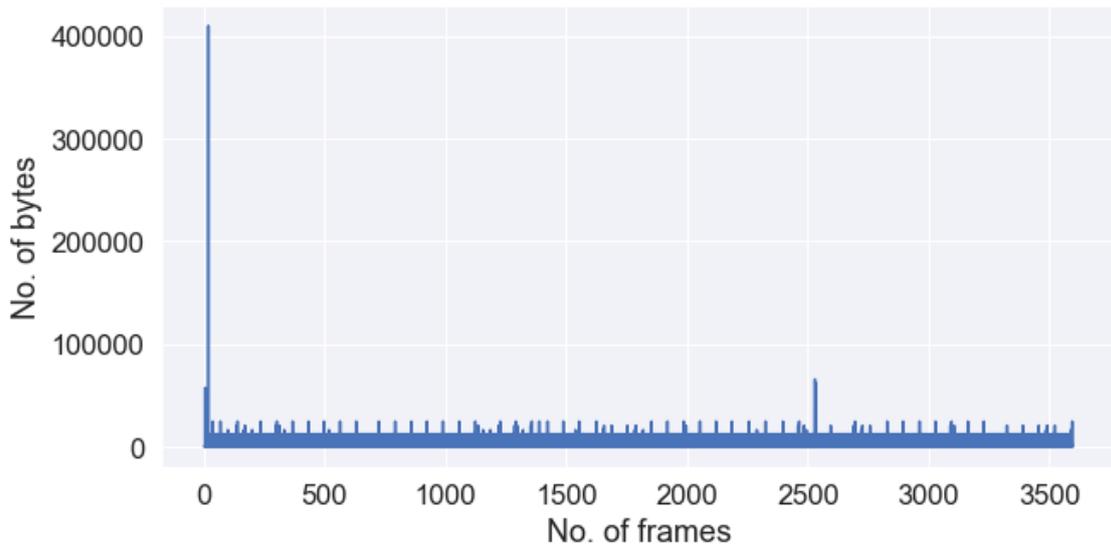


Figure 30b. Disk writes in S2.5.

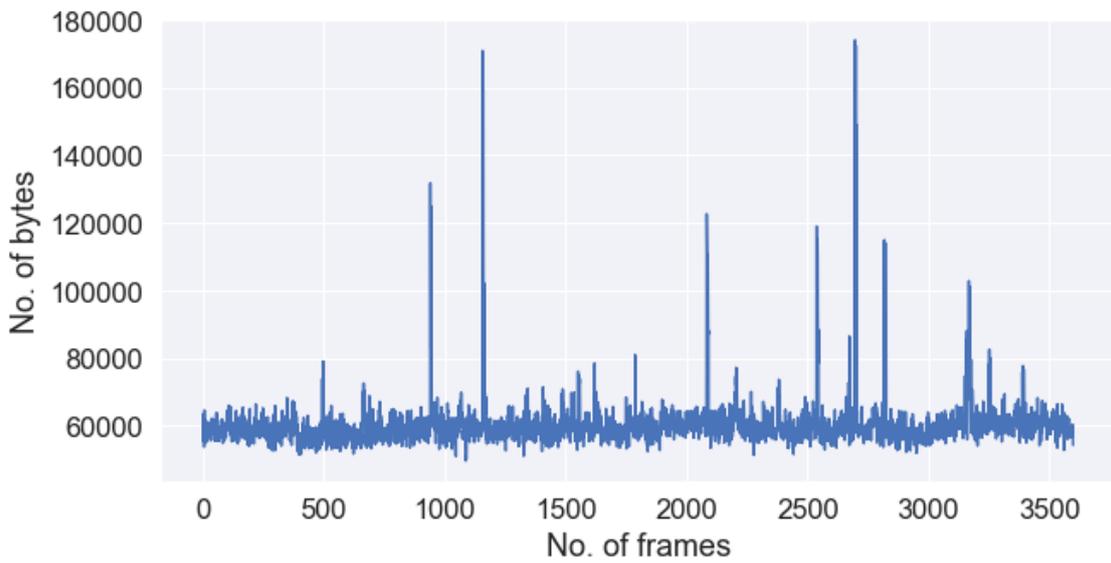


Figure 31a. Network bytes received in S2.5.

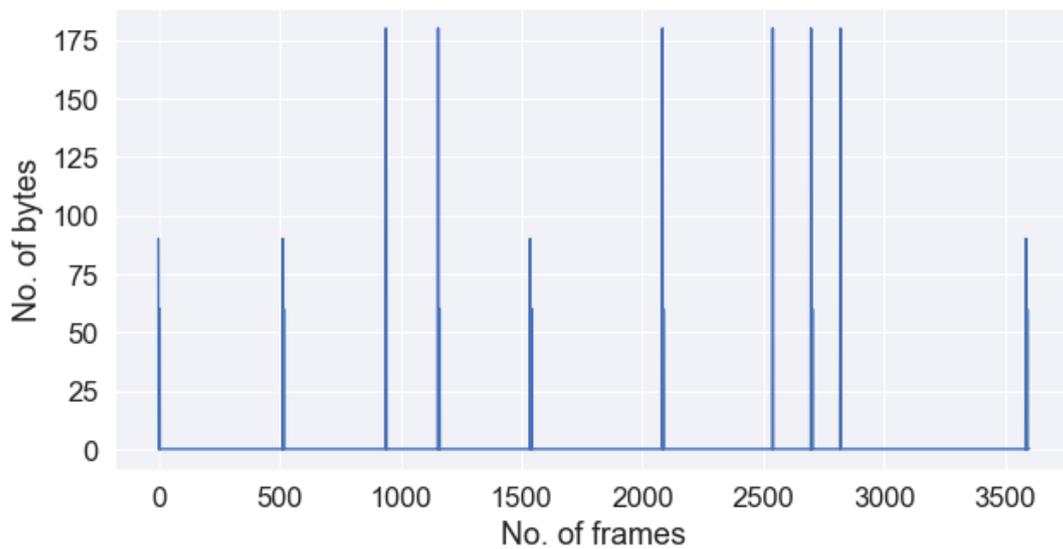


Figure 31b. Network bytes sent in S2.5.

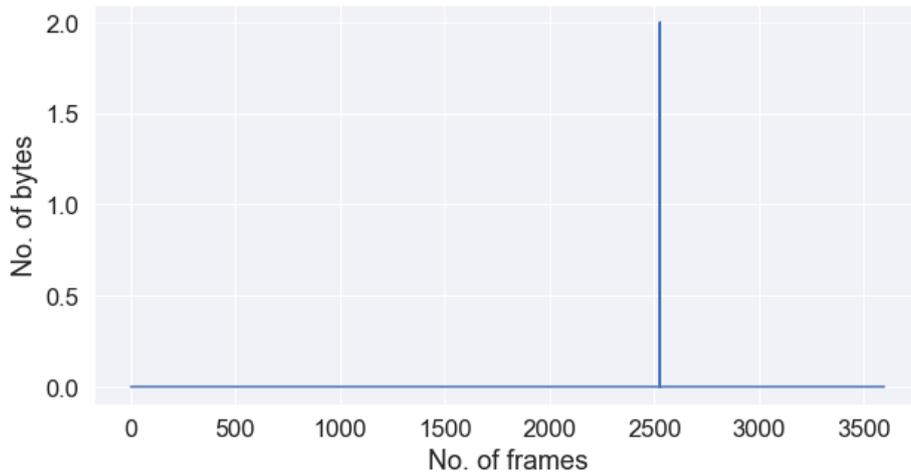


Figure 32a. I/O read requests S2.5.

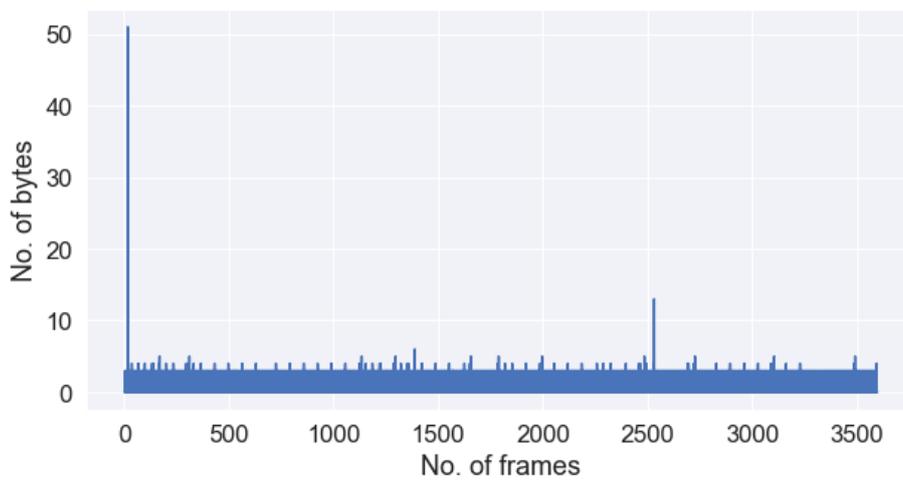


Figure 32b. I/O write requests S2.5.

The CPU utilization in VM with Ubuntu Desktop host operating system after installing OpenStack (S2.6) is presented in Figure 33. On x-axis, we can see number of frames and y-axis represents percentage of CPU utilized. The disk reads and writes after installing OpenStack are given in Figures 34a and 34b. On x-axis, number of frames are represented and y-axis represents data transferred in bytes. The network usage in terms of bytes received and sent after installing OpenStack are presented in Figures 35a and 35b. On x-axis, number of frames are being represented and y-axis represents data transferred in bytes. The number of I/O requests i.e., read requests and write requests after installing OpenStack are given in Figures 36a and 36b.

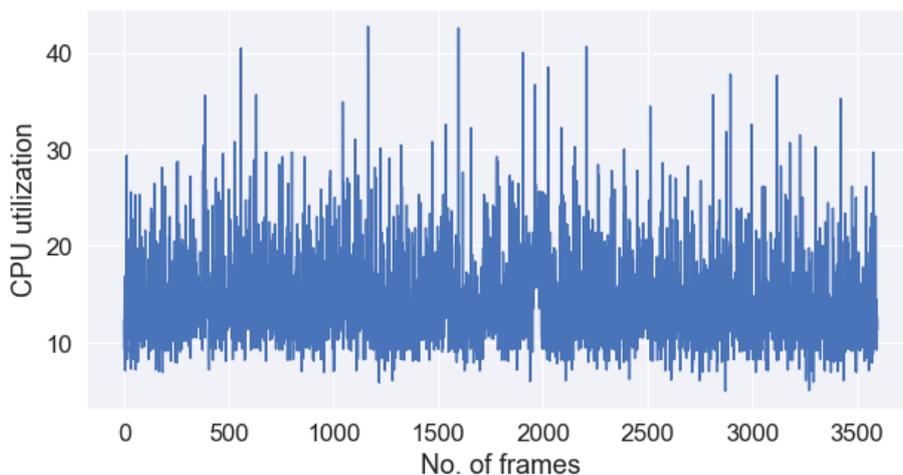


Figure 33. CPU utilization in S2.6.

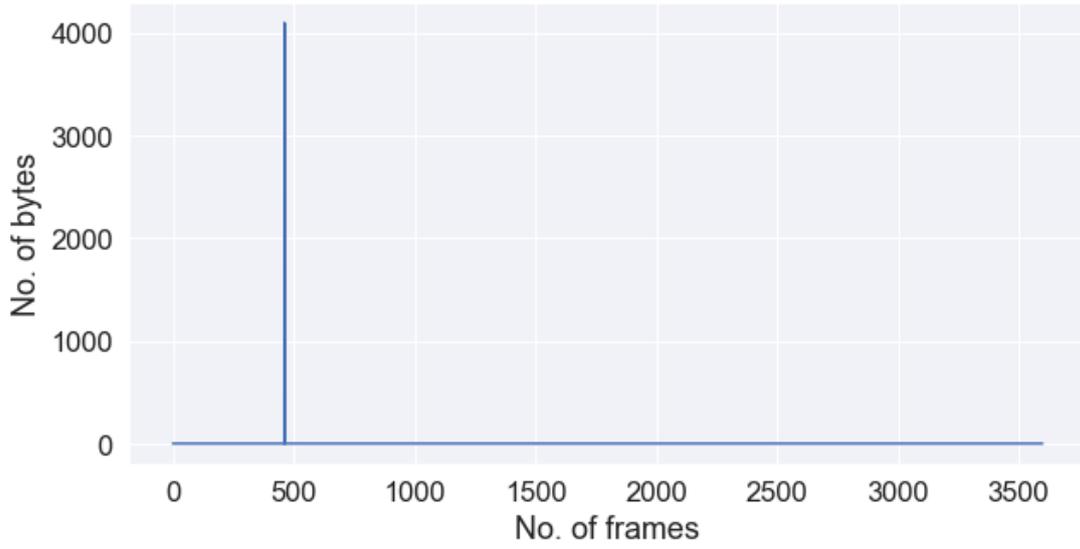


Figure 34a. Disk reads in S2.6.

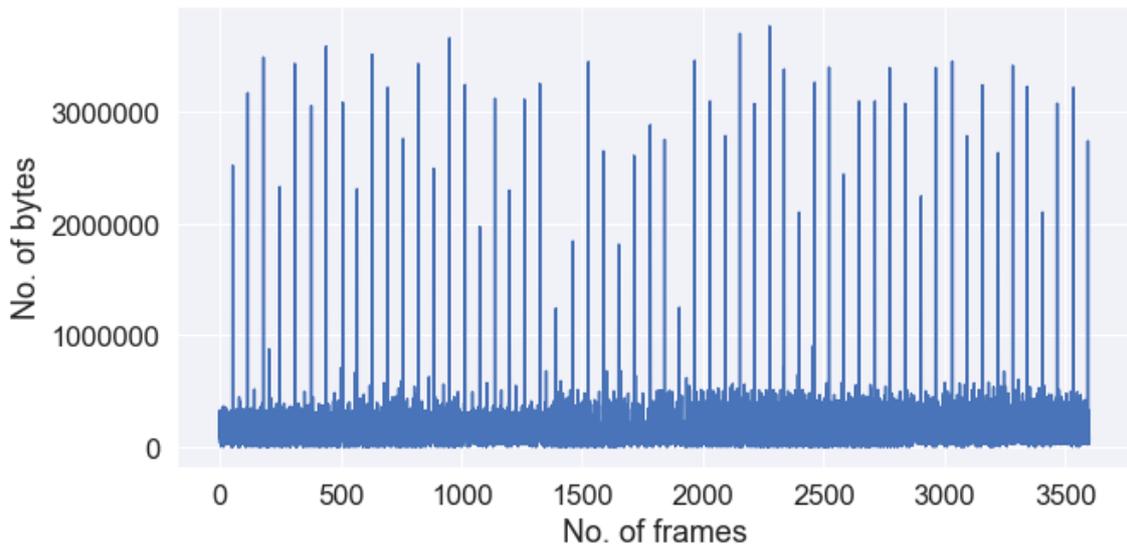


Figure 34b. Disk writes in S2.6.

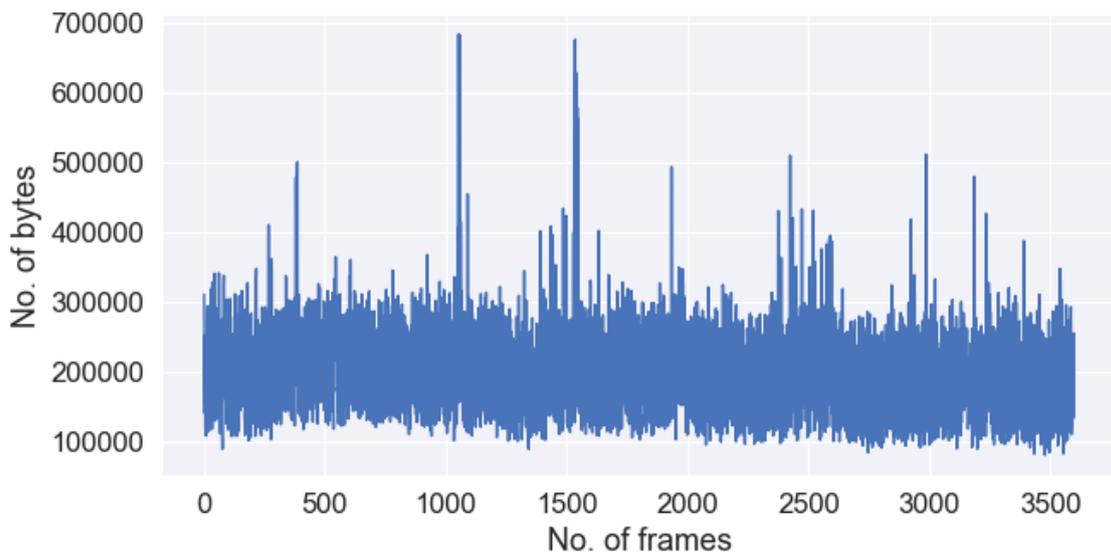


Figure 35a. Network bytes received in S2.6.

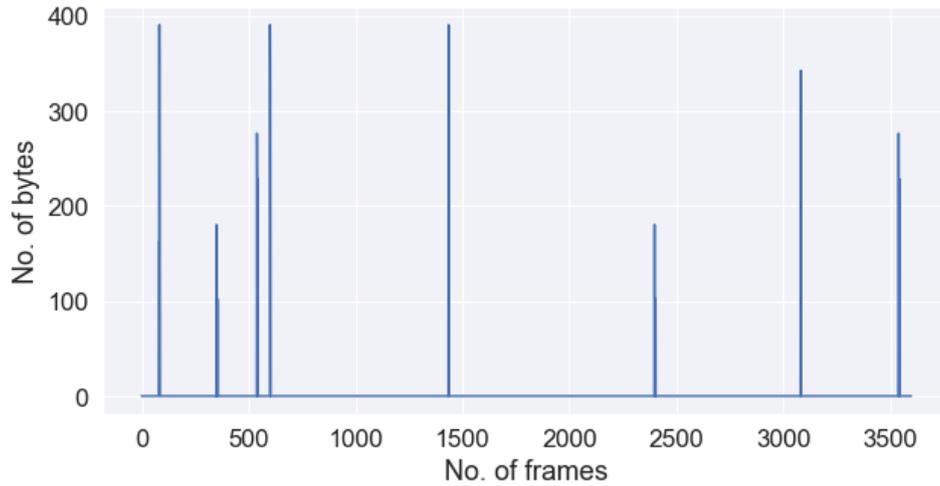


Figure 35b. Network bytes sent in S2.6.

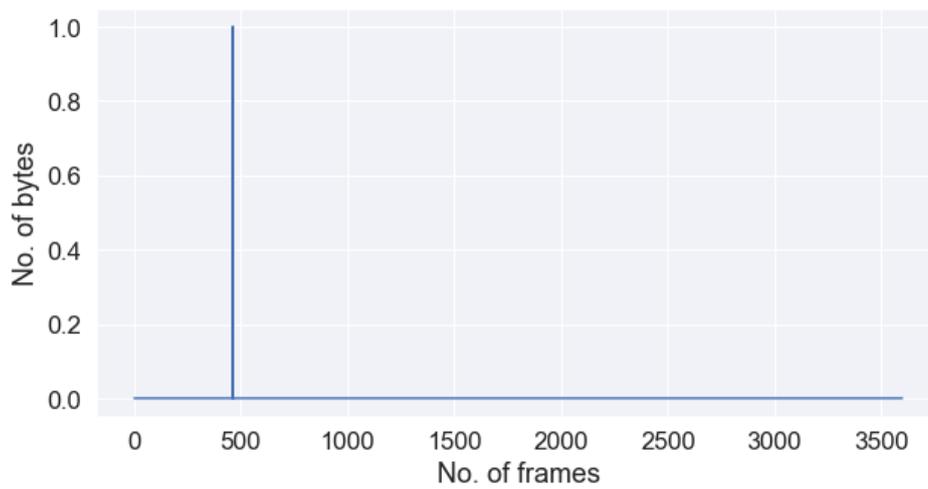


Figure 36a. I/O read requests S2.6.

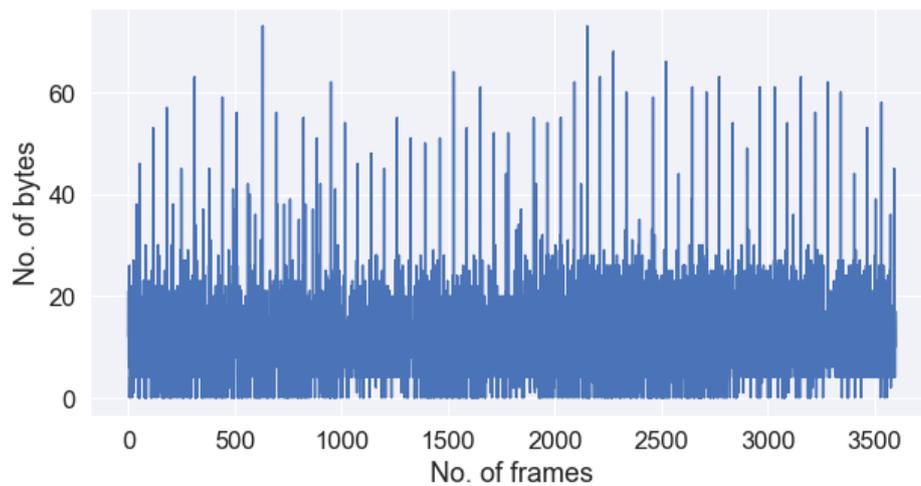


Figure 36b. I/O write requests S2.6.

The CPU utilization in standalone machine with Ubuntu Server OS is given in Figure 37. Similarly, CPU utilization in standalone machine with Ubuntu Desktop OS is given in Figure 38 and CPU utilization in Ubuntu VM is given in Figure 39. The overall comparison of CPU utilization in standalone machine with Ubuntu Server OS, Ubuntu Desktop OS and in Ubuntu VM before installing OpenStack is presented in Figure 40 and the comparison of CPU utilization in standalone machine with Ubuntu Server OS, Ubuntu Desktop OS and in Ubuntu VM after installing OpenStack is presented in Figure 41.

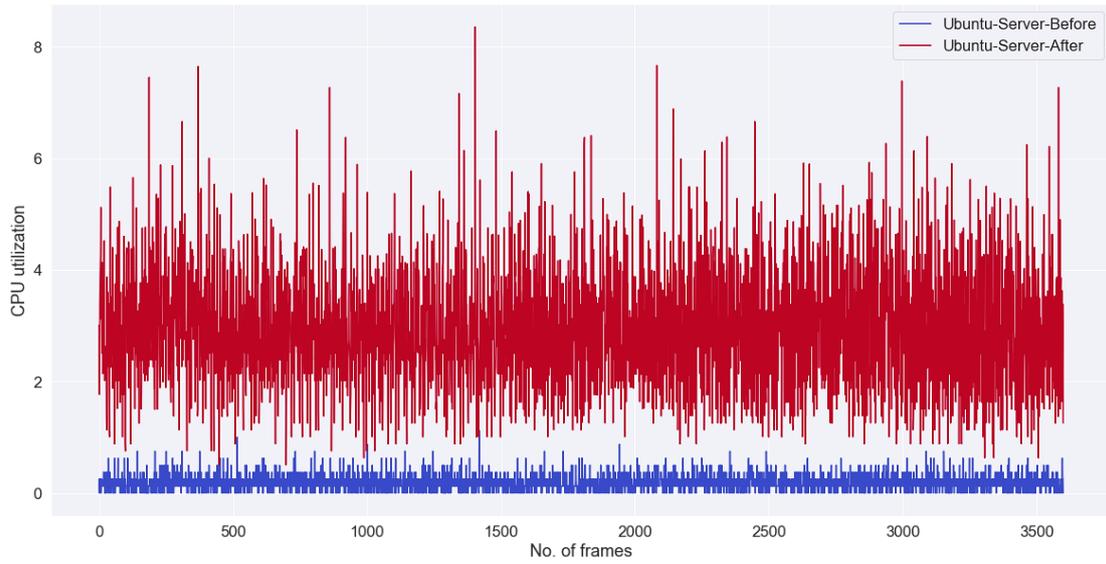


Figure 37. Ubuntu Server CPU utilization before and after installing OpenStack.

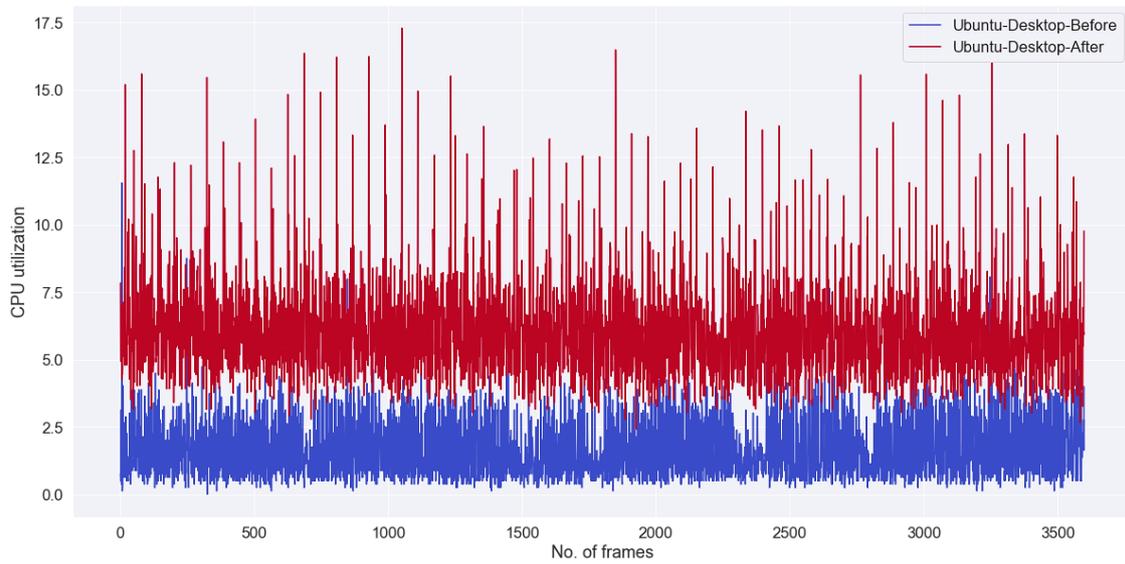


Figure 38. Ubuntu Desktop CPU utilization before and after installing OpenStack.

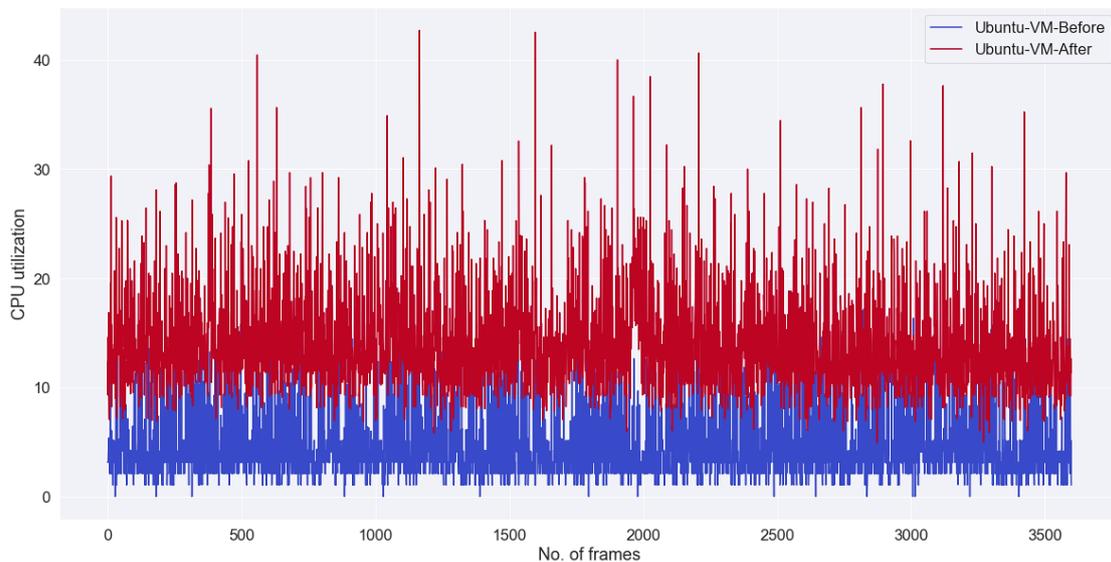


Figure 39. Ubuntu VM CPU utilization before and after installing OpenStack.

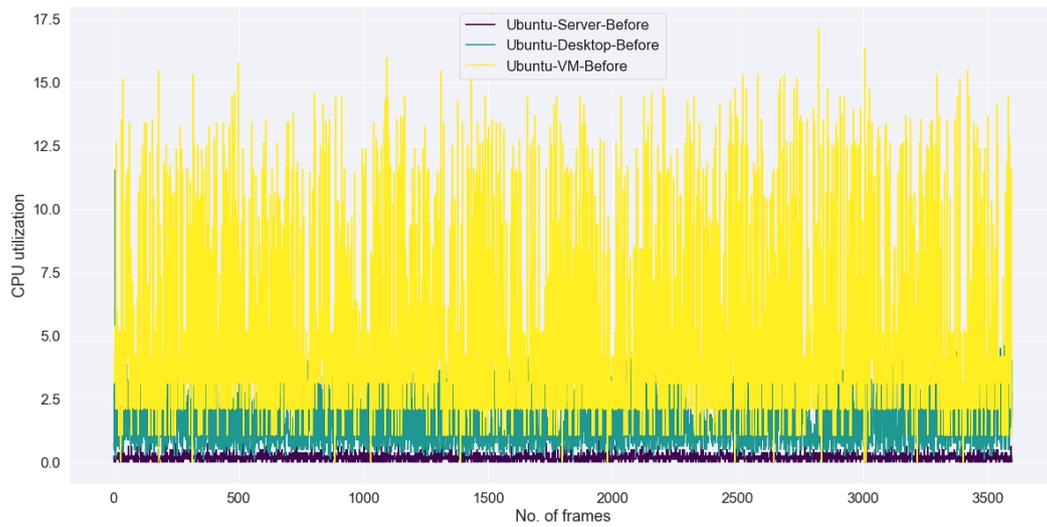


Figure 40. Ubuntu Server, Desktop, and VM CPU utilization before OpenStack installation.

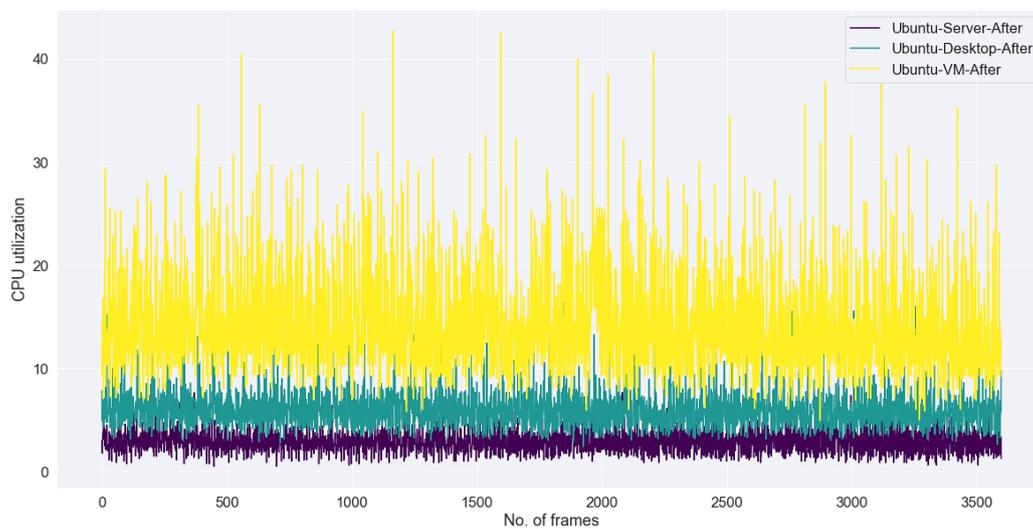


Figure 41. Ubuntu Server, Desktop, and VM CPU utilization after OpenStack installation.

4-3- Discussion

In scenario 1 (S1) based on the memory utilization given in Table 3, we can say that VM is consuming more memory than the stand-alone machine. That may be due to the inferior configuration of the machine on which the VM is installed. The CPU utilization is far better (less) in standalone machine than the VM before installing OpenStack as given in Figure 11. In Figure 12, after installing OpenStack, it is clear that most of the time VM is being heavily utilized (the peaks in the plot) by OpenStack services. Also we can see that while the CPU utilization in standalone machine is 3% on an average, it is 10% on average for the VM.

In scenario 2 (S2) based on the memory utilization given in Table 4, we can say that VM is consuming more memory than the standalone machine which is also observed in scenario 1 for different hardware. From Figures 37 to 39 we can say that there is significant impact on the machine after installing OpenStack. In the case of Ubuntu Server OS, there is nearly 2% increase in CPU utilization. In the case of Ubuntu Desktop OS, there is nearly 3-4% increase in CPU utilization. Finally, in the case of Ubuntu VM with Ubuntu Desktop host OS, there is nearly 10% increase in the CPU utilization. From Figure 40 we can see that even prior to OpenStack installation, the CPU utilization is significantly higher than those of other two cases and from Figure 41 the CPU utilization increased even more after installing OpenStack. Also, from the plots presented before, we can say that utilization of other resources like disk, network, and I/O requests increased significantly after installing OpenStack on the standalone machine for both types of operating systems.

Comparing S2.1 and S2.2, there is trivial difference in number of disk reads. Whereas, the number of disk writes increased significantly in S2.2. The number bytes received on the network also increased significantly in S2.2. There is trivial difference in number of bytes sent on the network. There is no significant difference in I/O read requests

which is not the case with I/O write requests. The amount of data sent in the form of I/O write requests increased significantly in the case of S2.2. Comparing S2.3 and S2.4, there is trivial difference in number of disk reads. Whereas, the number of disk writes increased significantly in S2.4. The number bytes received on the network also increased significantly in S2.4. In contrast to S2.2, the number of bytes sent on the network increased significantly in S2.4. There is no significant difference in I/O read requests which is not the case with I/O write requests. The amount of data sent in the form of I/O write requests increased significantly in the case of S2.4. Comparing S2.5 and S2.6, there is trivial difference in number of disk reads. Whereas, the number of disk writes increased significantly in S2.6. The number bytes received on the network also increased significantly in S2.6. There is trivial difference in number of bytes sent on the network. There is no significant difference in I/O read requests which is not the case with I/O write requests. The amount of data sent in the form of I/O write requests increased significantly in the case of S2.6.

Based on the results and discussion, it can be concluded that the performance of Ubuntu Server operating system on standalone machine is better than any other configuration. If possible, experimenting on a VM is discouraged as OpenStack has a significant effect on the host resources. When commodity hardware is used to setup a cloud, and the same hardware is shared by other people like in universities, it is suggested to use standalone machines for setting up the cloud. As for the host operating system, Ubuntu Server operating system offers significant benefit over Ubuntu Desktop operating system in terms of resource utilization. The work carried out in this paper is compared with other performance evaluation related work in the existing literature and the comparison result is shown in Table 5. This is the only work, which evaluated host resources utilization without creating any virtual machines in OpenStack.

Table 5. Comparison of existing cases in literature.

Author(s)	OpenStack Platform	Ubuntu OS	VMs Created in OpenStack	Host Resources Utilization
Sefraoui et al. (2012) [2]	√	X	√	X
Nasim and Kassler (2014) [3]	√	√	√	X
Gebreyohannes (2014) [4]	√	√	√	X
Xu and Yuan (2014) [5]	√	√	√	X
Mohammed and Kiran (2014) [6]	√	√	√	X
Grzonka et al. (2015) [7]	√	√	√	X
Raja and Rabinson (2016) [8]	√	√	√	X
Tornyai et al. (2016) [9]	√	√	√	X
Sha et al. (2015) [10]	√	X	√	X
Callegati et al. (2016) [11]	√	√	√	X
Shetty et al. (2017) [12]	√	X	√	X
Ahmad and Qazi (2018) [13]	√	√	√	X
Sajjad et al. (2018) [14]	√	√	√	X
Husain et al. (2020) [15]	√	√	√	X
Mandal et al. (2019) [16]	√	X	√	X
Saghir and Massod (2019) [17]	√	X	√	X
Figiela et al. (2018) [18]	X	X	√	X
Noertjahyana et al. (2019) [19]	√	X	√	X
Present research	√	√	X	√

5- Conclusion

The proliferation of cloud computing in industrial sector for resource provisioning has led to a profound increase in research and development activities in both industry and academia. Generally, many people in academia can't afford industry grade servers and other resources for testing cloud computing environment. They need a way to deploy and test cloud computing in an economic way. One of the popular ways for deploying cloud on commodity resources is OpenStack. The Devstack deployment allows a user to install OpenStack cloud on a single machine and test it. As OpenStack is quite complex, the host machine resources may be affected by it.

In this paper, the effect of OpenStack on underlying resources of the host machine is studied by conducting empirical analysis in various scenarios like different hardware configurations, different flavors of OSs and different virtualization levels. From the results it can be concluded that the best choice for deploying and testing an OpenStack cloud environment is to use a standalone machine with Ubuntu Server operating system. Unless you are compelled to do so, never use a VM for testing OpenStack cloud environment, as much of the resources are consumed by the OpenStack services which may lead to degraded performance later.

6- Acknowledgements

The author would like to acknowledge Vishnu Institute of Technology, India for providing the required hardware.

7- Conflict of Interest

The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

- [1] Suryateja, Pericherla S. "Experimental Analysis of OpenStack Effect on Host Resources Utilization." In *Advances in Intelligent Systems and Computing* (2020): 1059:11–19. Springer. doi:10.1007/978-981-15-0324-5_2.
- [2] Sefraoui, Omar, Mohammed Aissaoui, and Mohsine Eleuldj. "OpenStack: Toward an Open-Source Solution for Cloud Computing." *International Journal of Computer Applications* 55, no. 3 (October 20, 2012): 38–42. doi:10.5120/8738-2991.
- [3] Nasim, Robayet, and Andreas J. Kassler. "Deploying OpenStack: Virtual Infrastructure or Dedicated Hardware." In *Proceedings - IEEE 38th Annual International Computers, Software and Applications Conference Workshops, COMPSACW 2014* (September 18, 2014): 84–89. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/COMPSACW.2014.18.
- [4] Gebreyohannes, Merhawit Bereke. "Network Performance study on OpenStack Cloud Computing." Master's thesis, University of Oslo (2014).
- [5] Xu, Qifeng, and Jie Yuan. "A Study on Service Performance Evaluation of Openstack." In *Proceedings - 2014 9th International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA 2014* (January 20, 2014): 590–93. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/BWCCA.2014.120.
- [6] Mohammed, Bashir, and Mariam Kiran. "Experimental Report on Setting up a Cloud Computing Environment at the University of Bradford," December (December 15, 2014). <http://arxiv.org/abs/1412.4582>.
- [7] Grzonka, Daniel, Michał Szczygieł, Artur Bemasiwicz, Andrzej Wilczyński, and Marek Liszka. "Short Analysis of Implementation and Resource Utilization for the OpenStack Cloud Computing Platform." In *Proceedings - 29th European Conference on Modelling and Simulation, ECMS 2015* (2015): 608–14. European Council for Modelling and Simulation. doi:10.7148/2015-0608.
- [8] J. Beschi Raja, and K. Vivek Rabinson. "IaaS for Private and Public Cloud Using Openstack." *International Journal of Engineering Research And V5*, no. 04 (April 4, 2016). doi:10.17577/ijertv5is040191.
- [9] Tornyai, Roland, Tamas Pflanzner, Anita Schmidt, Balazs Gibizer, and Attila Kertesz. "Performance Analysis of an OpenStack Private Cloud." *Proceedings of the 6th International Conference on Cloud Computing and Services Science* (2016): 282–289. doi:10.5220/0005899502820289.
- [10] Sha, Leijie, Jie Ding, Xiao Chen, Xiaobin Zhang, Yun Zhang, and Yishi Zhao. "Performance Modeling of Openstack Cloud Computing Platform Using Performance Evaluation Process Algebra." *2015 International Conference on Cloud Computing and Big Data (CCBD)* (November 2015): 49–56. doi:10.1109/ccbd.2015.53.
- [11] Callegati, Franco, Walter Ceroni, and Chiara Contoli. "Virtual Networking Performance in OpenStack Platform for Network Function Virtualization." *Journal of Electrical and Computer Engineering* 2016, (2016). doi:10.1155/2016/5249421.
- [12] Shetty, Jyoti, Sahana Upadhaya, H. S. Rajarajeshwari, G. Shobha, and Jayant Chandra. "An Empirical Performance Evaluation of Docker Container, Openstack Virtual Machine and Bare Metal Server." *Indonesian Journal of Electrical Engineering and Computer Science* 7, no. 1 (July 1, 2017): 205–13. doi:10.11591/ijeecs.v7.i1.pp205-213.
- [13] Ahmad, Waqas, and Awais Salman Qazi. "Analysis of Interactive Utilization of CPU between Host and Guests in a Cloud Setup." *Computer Science and Engineering* 8, no. 1 (2018): 7–15. doi:10.5923/j.computer.20180801.02.
- [14] Sajjad, Muhammad, Arshad Ali, and Ahmad Salman Khan. "Performance Evaluation of Cloud Computing Resources." *International Journal of Advanced Computer Science and Applications* 9, no. 8 (2018): 187–99. doi:10.14569/ijacsa.2018.090824.
- [15] Husain, A, M H Zaki, and S Islam. "Performance Evaluation of Private Clouds: OpenStack vs Eucalyptus | *International Journal of Distributed and Cloud Computing-Volume 6 Issue 1*" n.d. Available online: <http://www.publishingindia.com/ijdcc/68/performance-evaluation-of-private-clouds-openstack-vs-eucalyptus/703/4908/> (accessed on 18 September 2020).
- [16] Mandal, Priyatosh, Rekha Jain, and Vaibhav Surwade. "Performance Analysis of OpenStack Controller." In *2019 International Conference on Smart Applications, Communications and Networking, SmartNets 2019* (December 1, 2019). Institute of Electrical and Electronics Engineers Inc. doi:10.1109/SmartNets48225.2019.9069800.

- [17] Saghir, Arsalan, and Tahir Masood. "Performance Evaluation of OpenStack Networking Technologies." In 2019 International Conference on Engineering and Emerging Technologies, ICEET 2019 (May 9, 2019). Institute of Electrical and Electronics Engineers Inc. doi:10.1109/CEET1.2019.8711829.
- [18] Figiela, Kamil, Adam Gajek, Adam Zima, Beata Obrok, and Maciej Malawski. "Performance Evaluation of Heterogeneous Cloud Functions." In Concurrency Computation. Vol. 30 (December 10, 2018). John Wiley and Sons Ltd. doi:10.1002/cpe.4792.
- [19] Noertjahyana, Agustinus, Henry Novianus Palit, and Daniel Kuntani. "Optimization of Computer Resources Using Openstack Private Cloud." In ACM International Conference Proceeding Series (September 20, 2019): 98–102. New York, New York, USA: Association for Computing Machinery. doi:10.1145/3362125.3362138.