A comparative study on docker containers and virtual machines

1st Bushra Tabassum
Bushra.tabassum@g.bracu.ac.bd

2ndMehedi
humaion.kabir.mehedi@g.bracu.ac.bd

3rd Sabbir
md.sabbir.hossain1@g.bracu.ac.bd

4th Annajiat
annajiat@bracu.ac.bd

***Abstract*—machine, either in the form of containers or cutting-edge hypervisors. The emerging technology can host apps.A widely used technological innovation in IT microservice businesses is server virtualization. A framework for newly presented services that run on a model of various tasks performed by smaller individuals is provided by virtualization.As a result, many operating systems on the cloud are physically required.It makes it easier to construct and deploy many virtual machines.on a single fundamental low-overhead method is evolving quickly.There are several lightweight microservice virtualization methods; docker has its machine in virtualization in the form of containers for hypervisors.An open-source platform is one of the emerging technologies that can host numerous apps.Oronix and Apache are benchmarks that use open-source software and measure CPU and memory performance.With the help of this technology, developers and system administrators can construct, produce, and measure the throughput, storage read/write performance, load tests, and operation speed. run programs on the Docker engine.Using common benchmark tools like Sysbench, Phoronix, and Apache benchmark, this article evaluates the performance of Docker containers and virtual machines with respect to CPU performance, Memory**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Introduction Industries discusses the emergence of virtualization technologies as Xen, HyperV, VMware vSphere, KVM, etc.Applications and IT dependencies can be deployed in bulk on a single virtual machine.The popularity of cloud computing applications has grown recently.The organized and the isolated produced many technology.Multiple apps can run on the same physical hardware thanks to virtualization. Virtualization solutions have a number of drawbacks, including enormous VM sizes, unstable performance from operating numerous VMs, a lengthy boot-up procedure, and inability to handle manageability, software upgrades, and continuous delivery issues.the introduction of a brand-new technique called containerization, which paved the way for virtualization at the OS level while bringing absorption to the hardware level. The hosts' operating system, which shares necessary libraries and resources, is used for containerization. Due to the lack of a guest OS, it is more effective. The processing of the application-specific binaries and libraries on the host kernel speeds up execution. The Docker platform, which mixes programs and their dependencies, aids in the formation of the containers. These containers are constantly running in a separate environment on top of the kernel of the operating system. By using Docker's containerization capability, the environment is guaranteed to support all pertinent applications [1].In this study, a number of experiments are conducted to measure and compare apps running on hypervisor-based virtual machines versus Docker containers. These tests enable us to comprehend the performance effects of containers and hypervisors, two important virtualization technologies. The paper is set out as follows; section 2 provides background information and a synopsis of various platforms and technologies. The mechanism used to comprehend performance comparison is covered in Section 3. The benchmarking results are shown in Section 4. Section 5 provides a conclusion and suggestions for future development.

Identify applicable funding agency here. If none, delete this.

II.

A. Docker

A technology known as containerization organizes system libraries, dependencies, and applications into a container-like structure. The applications which are built and organized can be executed and deployed as a container. Docker, a platform, ensures that the application functions in all environments. Additionally, it automates the deployment of apps into containers. The container environment in which the programs are run and virtualized is supplemented by Docker with an additional layer of deployment engine. Docker contributes to the creation of a rapid and light environment that can run code effectively. Docker's four key components are its containers, clients/servers, images, and engine. A thorough explanation of these elements will be provided in the following sections.

III. Docker engine

The Docker Engine, a client-server application that is installed on the host machine with the following components, is a crucial component of the Docker system. (A) Docker Daemon: A long-running software (the dockerd command) that aids in the development and execution of applications. (a) The Docker daemon is contacted via a Rest API. (c) To access the operations, a client sends a request to the Docker daemon through the terminal.

A. Docker client server

Two techniques can be used to create Docker Images. The main technique is to create an image using a read-only

template. The template comprises of base OS images, s as centos, Ubuntu 16.04, or Fedora, or any other lightwe base OS images. Every image is often built on a base im Every time a base image is built from scratch, a new in must be built. Making a change of this kind is referre as "committing a change." The following technique invo building a docker file that contains all the directions producing a docker image.The image will be produced all the requirements listed in the Docker file when the do build command is run from the terminal. This approac creating an image is referred to as automated.

B. Methodology

Utilizing benchmarking tools, KVM and Docker assessed in this part. The performance evaluation conducted using the benchmarking tools Sysbench [Phoronix [12], and Apache benchmark [13].These benchmarking tools assess the speed of operations as well as the performance of the CPU, memory, and storage.The two HP servers are used to compare performance across a range of parameters. One server is used as a virtual machine, running on top of the host OS (Windows 10), while the guest OS (Ubuntu 16.04) and docker engine are installed on top of the virtual machine. The other server is used as bare metal, running the host OS Ubuntu 16.04 and docker engine. On an HP server with two Intel Xenon E5-2620 v3 processors running at 2.40 GHz for a total of 12 cores and 64 GB RAM, all tests were run. All tests were conducted using the Linux kernel 3.10.0 and Ubuntu 16.04 64-bit.The same operating system, Ubuntu 16.04, was utilized as the base image for all Docker containers in order to ensure consistency and uniformity. The 12vCPUs and enough RAM are set up for VM. Fig. 3 displays the process for evaluating various.

C. Results

CPU performance :

A task's completion time or the number of operations the system has carried out in a specific amount of time (events/sec) are two ways to gauge computing performance. The server's virtual CPU core allocation has a significant impact on the outcomes. Sysbench, Phoronix, and Apache benchmark have all been used to test CPU performance comparison.

Maximum prime Number Operation : On the Sysbench tool test, which was done to determine how long it takes to process the largest prime number With an 80-second runtime and four thread operations, the operation's maximum prime number value is set at 70000. Figure 4 shows that a docker container executes an action significantly more quickly than a virtual machine. This is caused by the existence of the hypervisor in the virtual machine. As a result, the execution takes longer.

Memory performance : A cache and memory benchmarking tool called RAM speed/SMP (Symmetric Multiprocessing) is used to assess RAM speed for virtualization technologies like Docker and Virtual Machine. RAM Speed comparisons amongst virtualization systems are shown in Figure 6. When

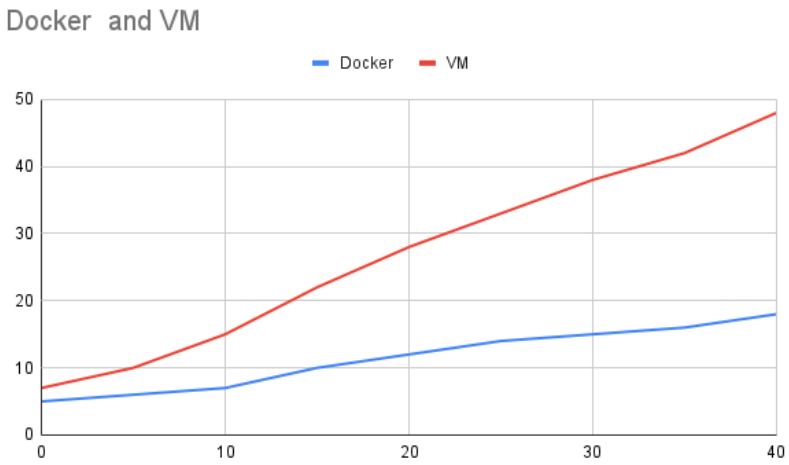

Fig. 1. Docker vs VM.

measuring the RAM speed, the two main key factors are taken into account. The RAM Speed SMP benchmarking program assesses the highest possible cache and memory performance while reading and writing individual blocks of data. It uses the INTmark and FLOATmark components.

INTmem and FLOATmem are artificial simulations that are closely aligned with the actual computing landscape. Each test has four subtests to assess various facets of memory function (Copy, Scale, Add, Triad). Data is transferred across memory locations using the copy command, which is (X = Y). Scale command, i.e. (X = n*Y), modifies data before writing by multiplying it by a specific constant number. When the command ADD is called, the data is first read from the first memory address and then from the second. The final step is to place the resulting data in third position (X = Y + Z). Add and Scale are combined in a triad.To scale the data, it is read from the first memory address, added from the second location, and then written to the third location (X = n*Y + Z).

Load testing : The Apache Benchmark tool, which calculates the maximum number of requests per second a given system can handle, is used to compare load testing performance. To test the load using the Apache Benchmarking tool, a Python application is run. Figure 8 demonstrates how the throughput analysis for VM is significantly lower than that for Docker. This is as a result of Virtual Machine having higher network latency than Docker. According to the analysis, a virtual machine cannot handle as many requests per second as a Docker container.

D. Conclusion

Conclusion: An developing lightweight virtualization technology is Docker Container. Docker containers and virtual machines are two virtualization technologies that are evaluated in this article. On virtual machine and Docker container-based hosts, performance is assessed in terms of CPU performance, memory throughput, disk I/O, load testing, and operation

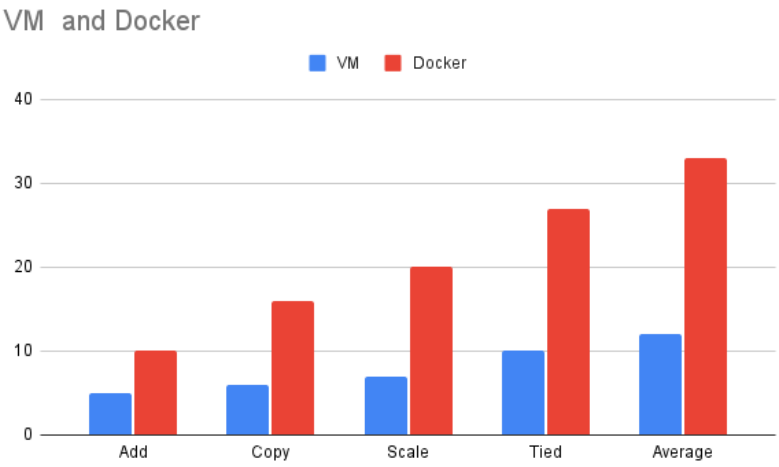

Fig. 2. Docker vs VM.

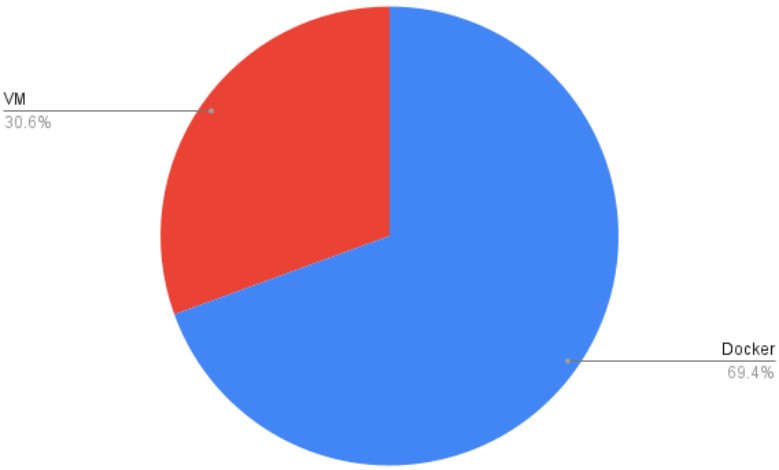

Fig. 3. Docker vs VM

speed measurement. In every test, it is seen that Docker containers outperform virtual machines, as the virtual machine is less effective due to the presence of the QEMU layer. Benchmarking tools like Sysbench, Phoronix, and Apache Benchmarking are used to evaluate the performance of both containers and virtual machines.

REFERENCES

[1] Docker. https://docs.docker.com/ 2019 [Online; Accessed 24-03-2019]
[2] Higgins J., Holmes V and Venters C. (2015) "Orchestrating Docker Containers in the HPC Environment". In: Kunkel J., Ludwig T. (eds)
[3] High Performance Computing. Lecture Notes in Computer Science, vol 9137.pp 506-513
[4] Sysbench. https://wiki.gentoo.org/wiki/Sysbench 2019 [Online; Accessed 24-03-2019]
[5] Apache Benchmark tool. https://www.tutorialspoint.com/apachebench/ 2019 [Online; Accessed 24-03-2019]