An Overview of Docker and an Evaluation of Its Performance

1st Md Sadman Hasan
Department of CSE
BRAC University
Dhaka, Bangladesh
md.sadman.hasan@g.bracu.ac.bd

2nd Fardin Rizvy Rahat
Department of CSE
BRAC University
Dhaka, Bangladesh
fardin.rizvy.rahat@g.bracu.ac.bd

3rd Md Badsha Faysal
Department of CSE
BRAC University
Dhaka, Bangladesh
md.badsha.faysal@g.bracu.ac.bd

4th K M Tahzeem Zaman
Department of CSE
BRAC University
Dhaka, Bangladesh
k.m.tahzeem.zaman@g.bracu.ac.bd

5th Md Sadiqul Islam Sakif
Department of CSE
BRAC University
Dhaka, Bangladesh
md.sadiqul.islam.sakif@g.bracu.ac.bd

6nd Md Sabbir Hossain
Department of CSE
BRAC University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

7nd Annajiat Alim Rasel
Department of CSE
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Docker includes some useful features for developers and administrators. Docker Engine is an open and free platform that allows users to create, distribute, and run applications in a mobile, compact environment and packaging tool. Docker Hub, a cloud-based app-sharing platform, is also available. Costs can be reduced by substituting a docker container for a standard virtual machine. It considerably reduces the cost of rebuilding the cloud development platform.**

***Index Terms*—Docker, Docker Container, Hypervisor, Containerization, Virtual Machine(VM).**

I. Introduction

Docker is an open source framework for executing the application and streamlining the development and delivery processes. Docker-based programs are bundled with all essential dependencies within a container, which is a regulated format. These containers continue to perform in seclusion atop of the kernel of the operating system. The added layer of abstraction could affect performance. A thorough literature study will be used to examine the performance of Docker technology in this research. The sections that follow will go over Docker technology, as well as a more extensive discussion of Docker and its components. Furthermore, the paper will include a brief comparison of Virtual Machine and Docker technology. Finally, the study will conduct a thorough examination of the advantages and disadvantages of docker containers.

II. Docker

Docker is a free and open-source containerization technology that was first released in 2013 and has swiftly grown in popularity due to its comprehensiveness and simplicity of use as pointed out by Raho et al. (2015) [6]. It allows programmers to bundle programs into containers, which are conventional executable elements that combine application source code with the OS libraries and components needed to operate the program in any environment. Containers make it easier to install scattered apps and are becoming more prevalent as enterprises transition to cloud-native development and hybrid multi-cloud configurations. Docker is a toolset that enables developers to build, deploy, operate, upgrade, and stop containers with basic instructions and labor-saving automation via a single API. Developers may create containers without Docker, however the platform simplifies, simplifies, and secures container creation, installation, and administration [5].

III. How Docker works

Docker is comprised of four major components: Docker Client and Server, Docker Images, Docker Registries, and Docker Containers. The following is the official high-level Docker architectural diagram, which depicts the standard Docker process:

IV. Docker Daemon and Client

Docker is built on a client-server model as is shown in figure 1. The Docker daemon is responsible for all container-related operations. The Docker client sends commands to the daemon via the CLI or REST API [4]. The Docker client can run on

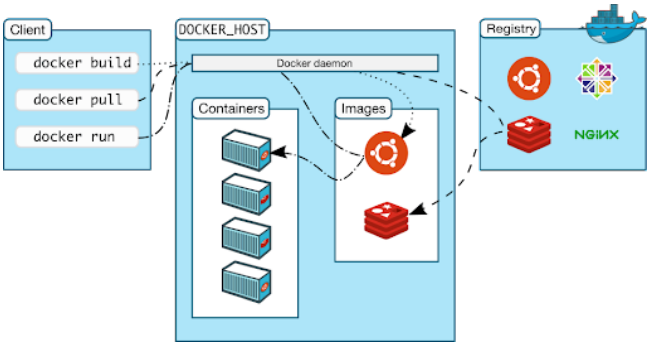

Fig. 1. Docker architecture. [3]

the same host as the daemon or on any other host. The Docker client also has the ability to communicate with many daemons.

V. Docker Images

Docker images provide executable application source code as well as the tools, libraries, and dependencies required for the application code to execute in a container. When you execute the Docker image, it creates a single (or many) container instances. Docker images are made using layers, and each layer represents a different image version. A new top layer is produced whenever a developer makes modifications to the picture, and it replaces the old top layer as the current version of the image. Previous layers are kept in case of rollbacks or reuse in other projects. Since numerous live container instances may operate from a single base image and share a stack, this iterative image-creation process improves overall efficiency [5].

VI. Docker registry

It's a repository for Docker images. The Docker registry allows you to share photos. It is used to store Docker images in a central location. The registry allows us to designate image versions in repositories to keep track of them. A register might be made public or kept private. Docker Hub is a Docker Inc.-provided hosted registry service. It allows us to upload and download photos from one convenient spot [4].

VII. Docker Container

Docker containers are live instances of Docker images that are currently executing. Containers are live, ephemeral, executable content, whereas Docker images are read-only files. Users may interact with them, and administrators can use Docker commands to change their settings and circumstances [5].

VIII. Hypervisor

A hypervisor is a delicate layer of programming that supplies apparatus attention to working frameworks by allowing numerous working frameworks or versions of the same working framework, acknowledged to as guests, to execute on a single server Computer. A bunch of Hypervisor variants are utilized in virtualization and superior execution registering, each based on their reasonable attributes and execution measurements. A similar investigation of these assortments will be displayed beneath.

IX. Type 1 Hypervisor

Because it functions straight on top of the fundamental structure, it is also known as a "local hypervisor" or "uncovered metal," as seen in Fig. 2. VMM is a small block of software that plans and allocates framework assets to virtual servers since there is no functioning framework operating behind it. VMware ESX and Xen are two instances of Type I VMM. The VMM in the present circumstance gives gadget drivers that the visitor OS uses to get to the fundamental equipment straightforwardly as seen in the examination by Galvin B et al, 2009. [2].

X. Type 2 Hypervisor

This Hypervisor is otherwise called facilitated VMM since it runs as an application on a normal working framework called the host working framework. The host OS is totally uninformed about Type II VMM and deals with it like some other interaction. It for the most part handles I/O for the visitor OS. The visiting Operating system delivers the I/O request, which would be picked up by the host Operating system, which therefore passes it to the device driver, which performs the I/O. The fulfilled I/O demand is sent back towards the visiting OS via the host OS, as demonstrated by Goodman et al, 1987. [12].

XI. Hyper V

Windows hyper-v is a hypervisor-based virtualization system for x86-64 architecture. Customers may virtualize their infrastructure just to save cost while using this safe virtualization platform with incorporated administration. In Microsoft Hyper-V, the hypervisor's primary job is to generate a separated runtime environment termed as a partition. A partition is a logical model that houses the operating system. The hypervisor must have one root or parent partition that runs Windows Server 2008 and afterwards produces child partitions that host the guest os using the hypercall application programming interface. A hypervisor should not only establish virtual machine instances with instruction-set architecture that is invisible from physical hardware, but it should also do it swiftly. A virtual machine ought to be capable of executing operations at a pace comparable to hardware. For most situations, productivity is accomplished by the use of a dual-mode processor architecture, in which supervisor mode enables both privileged and unprivileged commands to be run, whereas user mode only enables non-privileged commands to be processed. The hypervisor assumes charge of every inbound interruptions or traps while in supervisor mode. As claimed and proved by Thomas C Bressoud et al, 1996, all those other software, including the virtual machine's operating system kernel, operates in user mode. [11]

Comparison

Hypervisor	Host OS	Guest OS	Type of Hypervisor
Microsoft Hyper-V	Windows 2008 w/Hyper-v Role, Windows Hyper-V Server	Windows, SUSE	1
KVN	Linux	Linux, Windows	1-2
XEN	Linux, Solaris, NetBSD	BSD, Linux, Solaris, Windows	1
VMWare ESX	None Bare-metal	Windows, Linus, Novell, Netware,Sun Solaris, FreeBSD	1

Fig. 2. Comparison

XII. Takeaway

The Hypervisor is a software technology that allows multiple operating systems to run on the same physical hardware, or virtualization. Based on the taxonomy presented in this work, researchers can properly choose the correct hypervisor. This article also contains data that may be used to assess existing hypervisor solutions.

XIII. Docker vs Hypervisor

Creating, simulating a computing environment which is abstracted from the physical computing hardware and basically creating a computer from a computer virtually is computer virtualization. And again can be created multiple virtual computing instances from the hardware and software components of a single machine. These virtual instances can be a computer with traditional sense or can be used as a storage repository, application, server, or networking configuration. [9] And Hypervisor is an application that enables virtualizations, which is very lightweight, sits between the physical layer of hardware, and allows it to operate multiple OS isolated. A hypervisor acts as a mediator, directing resources from your infrastructure's raw materials to the numerous computer instances. On the other hand, Containers like docker just virtualize the OS, sit on the top of the physical servers, and use host kernel, binaries, and libraries. Sharing the host OS resources, for example libraries, binaries. A server may execute various workloads that can be handled by a single installation of the operating system. Containers are much lighter than hypervisor based virtualization. They are only a few megabytes in size and start-up in seconds. It makes a bigger impact on the development section, as it takes less space to isolate and uses fewer resources with faster output through the containerizing. [7]

Docker is a Linux container-related tool. Linux containers are a virtualization approach that allows numerous apps to operate on a single server. The Dot Cloud discovers this utility. The client receives commands from the user and communicates bidirectionally with the daemon. The Docker daemon is running on the host computer. If another component requests services from the daemon, the daemon acknowledges it. The user cannot connect directly to the daemon; it must first connect to the docker client and then to the user. [9]

Advantage and disadvantage of docker compared to VM: It is not possible to compare docker against hypervisor head to head. The way they work is different. They have their advantages and disadvantages over one another. We can not just replace virtual machines with docker. Hypervisor helps us to break down a large computer's resources into chunks, depending on the needs of the user. The question arises when using docker provides more efficiency and ease of use compared to VM's. Docker provides flexibility speed in terms of deployment. For example, if a user wants to deploy a PHP-based application for testing, the user needs to install PHP, install apache or Nginx as an HTTP server, or install a database server. All these installations take time and it is not a simple process either. All of these tasks can be minimized by implementing docker. It takes minimal commands to pull an image from the repository and test the application [10]. Not to mention, if something goes wrong during the setup process it is very easy to start from the beginning compared to working on a virtual machine. The same applies to the production use case. Docker helps developers deploy new applications or to modify or fix bugs easily. A lot of cloud services depend on Docker's concept. Such as Kubernetes and FAAS [11]. Up until now, we only discussed how docker is a better way of deployment than a virtual machine, but a great advantage of docker over VM is performance and efficiency. Virtual machines are heavy as they all require to run individual kernels and basic services of the OS. Docker acts as if it is an isolated environment but they share the kernel with the host OS and even share parts of the image with other images. It uses a protocol named fs layer. On the same hardware, users can achieve more computing power by using docker. But when it comes to the data center's resource distribution, the hypervisor is needed.

XIV. Performance Comparison

We ran some tests to compare the performance of docker vs VM. We compared CPU performance, memory performance, disk I/O and how they handle the various kinds of load [8]

XV. Methodology

For our test, we chose KVM as our hypervisor. We used Sysbench, Phoronix, SMP, IOzone, and Apache Benchmark as benchmark tools. These tools measure CPU, memory, and Disk I/O. Ubuntu 20.04 is our OS of choice for these tests.

XVI. CPU Performance

Sysbench tool measures CPU performance. A maximum prime number of 50000 was taken and measured its perfor-

Docker	Hypervisor
lightweight	Heavyweight
Native performance	Limited Performance
VM's in hypervisor needs its own OS	Containers shares their host OS
Faster startup(milliseconds)	Slower startup(few minutes)
Use less memory	Need memory like an actual rig
Process level isolated	Fully isolated
Possibly less secure	More secure in networks
Rather than moving can be created and destroyed easily	Easy to move to a new host
Lots of prebuilt images	Hard to find and manage

Fig. 3. Performance Comparison

mance in 60 seconds. The result shows docker is much faster than VM.

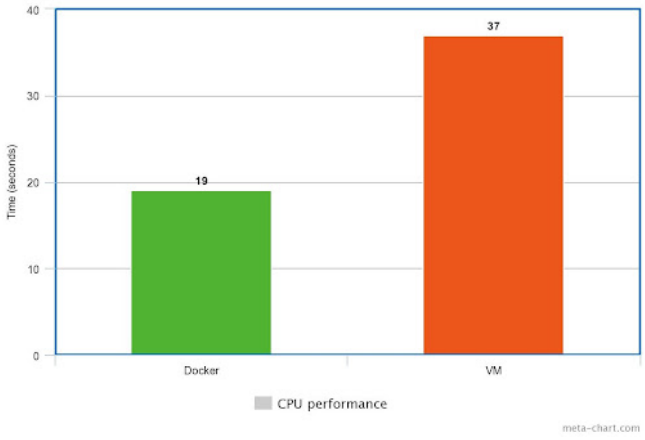

Fig. 4. CPU Performance

XVII. Memory Performance

SMP benchmark tool was used to benchmark memory performance. It shows that Add, Copy, Scale, Triad, and Average had a higher throughput on docker.

XVIII. Disk I/O Performance

IOzone was used to benchmark I/O throughput. In both read and write docker was faster than VM.

XIX. Load Testing

Apache Benchmark measures The number of requests a server can process in a certain amount of time. Here a small flask web server was being tested on docker and VM. Here the result shows, Docker can handle more request per seconds than VM.

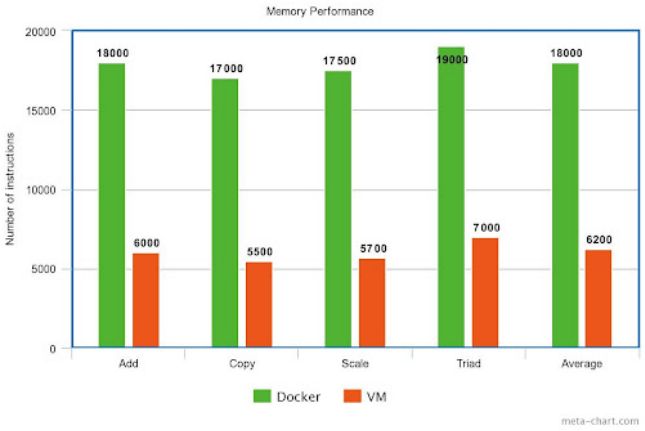

Fig. 5. Memory Performance

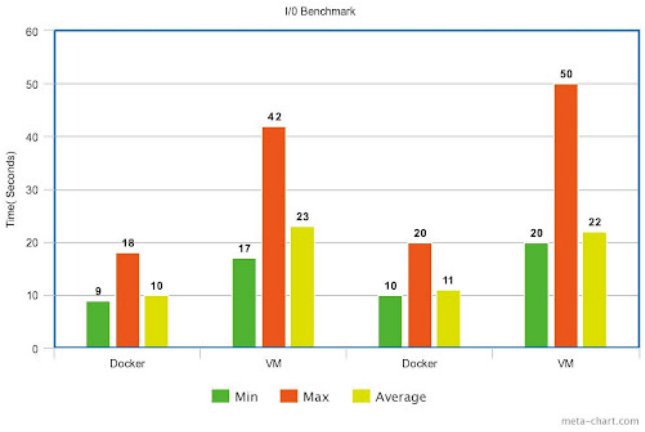

Fig. 6. Disk I/O Performance

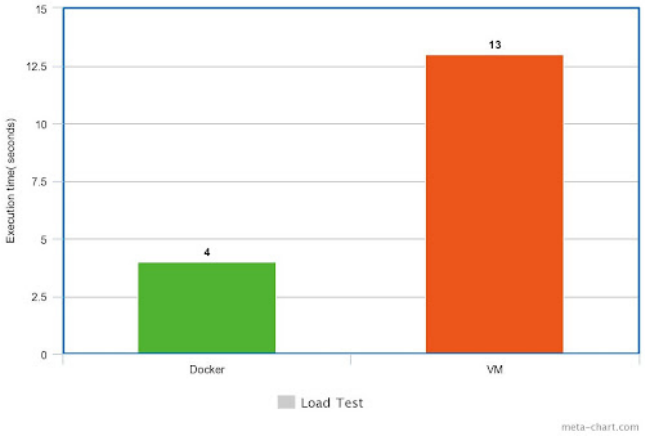

Fig. 7. Load Testing

XX. Conclusion

Docker Container is a new lightweight virtualization solution that is fast gaining traction. This study analyzes and contrasts two types of virtualization technologies: docker containers and virtual machines. CPU efficiency, memory through-

put, disk I/O, load testing, and execution speed measurement are all conducted on virtual machine and Docker container-based hosts. Docker containers outperform VMs in every test because the existence of the QEMU layer inside the vm leads it to be less economical than Docker containers. Comparing tools including such Sysbench, Phoronix, and Apache benchmarking are being used to monitor the performance of containers and virtual machines. We want to work upon container scheduling in Docker in the future, as well as a more safe variation of containers that will remove security limitations.

REFERENCES

[1] Desai, Oza, Sharma, Patel, A. D. R. P. S. B. P. (2013). Hypervisor: A Survey on Concepts and Taxonomy. Hypervisor: A Survey on Concepts and Taxonomy, 2(3), 1–4. Hypervisor: A Survey on Concepts and Taxonomy

[2] P, Galvin B. VMware vSphere Vs. Microsoft Hyper-V: A Technical Analysis. [White Paper] s.l. : CTI Strategy, 2009

[3] Docker overview. (2022, January 14). Docker Documentation. https://docs.docker.com/get-started/overview/

[4] DevopsCube. (2020, September 13). What Is Docker? How Does It Work? https://devopscube.com/what-is-docke/

[5] IBM Cloud Education. (2021, August 4). Docker. https://www.ibm.com/sg-en/cloud/learn/docker

[6] Raho, M., Spyridakis, A., Paolino, M., Raho, D. (2015, November). KVM, Xen and Docker: A performance analysis for ARM based NFV and cloud computing. In 2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE) (pp. 1-8). IEEE.

[7] Eder, Hypervisor- vs. Container-based Virtualization https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2016-07-1/NET-2016-07-1_01.pdf

[8] Potdar, Narayan, Kengond, Mulla, Performance Evaluation of Docker Container and Virtual Machine,https://www.researchgate.net/publication/341907958_Performance_Evaluation_of_Docker_Container_and_Virtual_Machine

[9] P and Suresh, Comparative Analysis on Docker and Virtual Machine in Cloud Computing, https://acadpubl.eu/jsi/2017-117-7/articles/7/19.pdf

[10] Cheng J(June 18,2021). PHP Websites using Docker Containers with PHP Apache andMySQL. https://www.section.io/engineering-education/dockerized-php-apache-and-mysql-container-development-environment/

[11] GB .P(27 April,2018) OpenFaaS on Docker. https://medium.com/@pavithra_38952/openfaas-on-docker-440541d635a2

[12] Bressoud, Schneider, T. C. B. F. B. S. (1996). Hypervisor-Based Fault-Tolerance. Hypervisor-Based Fault-Tolerance, 1–28. http://roc.cs.berkeley.edu/294fall01/readings/bressoud.pdf

[13] BERNSTEIN,P.A.,HADZILACOS,V.,ANDGOODMAN,N.1987.Concurrency Control and Recovery in Database Systems. Addison-Wesley, Reading, Mass.