International Journal of Pure and Applied Mathematics
Volume 117 No. 7 2017, 175-184
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Comparative Analysis on Docker and Virtual Machine in Cloud Computing

Prakash P[1] and Raghavi Suresh[2]
Dept. of Computer Science and Engineering
Amrita University
Coimbatore
India.

Abstract

In Cloud computing, Virtual machine is a software program that helps in running applications and programs which permit workloads to be isolated from one another and for the resource usage is controlled. However, the extra levels of abstraction involved in virtualization reduce workload performance, which results in worse performance. Meanwhile in virtual machine performance degradation is high because of heavy load in process, so slow provisioning happens in resources. Newer advances in container-based virtualization simplify the deployment of applications while continuing to permit control of the resources allocated to different applications. Here, all related applications are packaged into a single unit. So it is easy to migrate from one place to another. Containers are light weight processes which separate into files and directories to form a single coherent file system. So it can dynamically manage the workload and can be easily scale up or scale down the resources. In our project, we explore the performance of traditional virtual machine deployments, and contrast them with the use of Docker containers. Evaluations are made on factors like CPU stress, memory, and booting up processes.

Keywords*: Docker Containers, Virtual Machine, Docker Hub/Registry, Docker Images, Docker File, Hypervisor, Cloud.*

.

1. Introduction

Nowadays cloud computing is omnipresent. Cloud computing is also known as on-demand computing service over the internet, where resources and data are shared rather than on a local server or a personal computer. It is a model for enabling ubiquitous network access shared to a pool of configurable computing resources. It follows a pay as you go process where it lets users pay for what they use [1]. Normally in traditional computing applications run on a hardware and software. In this case processes might stay idle in most of the times, so efficiency is low. Therefore the requirements cannot be easily changed or distributed in the system. To overcome the underutilization issues we move to a process called virtualization.

1.1 Virtualization in Cloud

Computer virtualization is a method of partitioning a computer's hardware and in return presents multiple virtual machines on the computer platform providing the ability to run multiple operating systems and configurations without requiring multiple stand-alone computer systems as discussed by Jason Latham [2].Virtualization is a key technology to combine the collection of servers, so it optimizes the resources and reduce the managing cost. Data centre networks has more profound impact on virtualization. It allows running multiple heterogeneous operating systems concurrently in a single host machine, so easy to share the CPU, memory resources. It is not just the cost but the systems isolation is also improved. Virtualization can be made into a single unit by encapsulating applications and operating systems. So the files can be moved and copied from one place to another and the hardware is utilised properly. Even if one virtual machine fails, we can restart the other virtualized server and the necessary services can be used from the resource pool. Multiple virtual servers are originated where virtualization is accomplished. Creating abstract layer of system resources in virtualization allows reducing the complexity of hardware and software. Hypervisor is called as virtual machine manager (VMM) that is usually used software for virtualization. It allocates resources to each virtual machine it creates and it also handles the deletion of virtual machine says Joseph M.Kizza [3]. This mechanism manages all

guest operating system in a hassle-free manner. The various applications run directly on the hardware with the help of virtual partitioning capabilities.

1.2 Categories of Virtualization

1.2.1 Software Virtualization

Software virtualization is also renowned as application virtualization. It is the software technology that can be installed locally within a single unit, and manages the interaction between the application components. In addition there are two types of software virtualization and they are remote and streaming. Remote applications are an efficient way that runs on the server. Users can interact with their application over network via remote protocol. Application can be run on its own instance that can be shared with other users. Streaming application is executed on the end users local computer without using network. It will not execute on the end users local computer without using network. It will not interfere with other application, and hence, will achieve isolation.

1.2.2 Desktop Virtualization

Desktop virtualization is also known as client virtualization. The client machine is virtually monitored by the system administrator. This centralised management enhances the security and the client management. There are three levels of client virtualization remote, local and application. The remote level is also known as server hosted virtualization. It is treated on a server and operates the client through the network. The local level is also known as client hosted virtualization that runs on a local machine. The application level enables the applications to run on the isolated environment [4].

1.2.3 Storage Virtualization

Storage abstraction is the process of partitioning physical storage into logical storage. It enhances the utilization of space and performance there are three types DAS, NAS and SAN. DAS is Direct Attached Storage that is directly connected to server. NAS is Network Attached Storage that will connect to the network and it shares resources such as file sharing. SAN is Storage Area Network that connect different servers through networks. Storage virtualization spans the storage capacity automatically. So it reduces manual provisioning [6].

1.2.4 Network Virtualization

Network virtualization is used for monitoring and managing an entire network as a single entity. Many resources such as network servers and services are pulled into single pool which cannot be correlated with physical components. Physical networks can be separated into the virtual network or can be combined to local network to share the resources. Network administrator groups the machines and their traffic logically. It will utilize the physical infrastructure efficiently and improve the network productivity.

1.2.5 Server Virtualization

Server virtualization is also known as hardware virtualization. The hardware of one physical server is divided into multiple isolated virtual environments which uses the software that is called hypervisor that abstracts the significance of physical server. The guest operating system can run without making any alterations. The guests can be deployed with different operating systems because the guest is operating completely independent of the OS of host and is not aware that it is not using real hardware.

Virtualization has various advantages like the isolation because multiple operating systems can run on a same physical machine simultaneously. So there is easy maintenance, availability and recovery. This may involve certain difficulties such as

- Performance degradation due to the heavyweight of virtual machine
- Managing and migrating virtual resources are difficult
- Hypervisor controls all guest operating system as the OS instance increases managing and also hard to control as it will lead to security issues.
- Lack of application portability

To overcome these drawbacks we are moving to docker.

1.3 Docker

Docker is an open platform for developers and system administrators to be build, ship and run distributed applications [7]. So anyone can contribute to docker and extend it to meet their own needs. Docker allows packaging an application with all of its dependencies into a standardized unit for software development [7]. Docker's configuration mechanism is light-weighted because it uses UFS (Union File System) which allows files and directories of separate file system to be formed into a single coherent file system. Code-pipeline Management provides consistent environment to the application from developer through production making the code easy for development and deploying it. Development environment has a very low memory because it adds overhead of working remotely whereas docker easily allows different services to run within itself [12].

In docker various applications can deploy and run securely isolated in containers. The isolation and security allows to run many containers parallely on the host. Application isolation allows the ability for multiple servers to save on cost with the ability to share unused memory across the instance. Docker provides many tools that work well with containers. It includes checkpoint container and container versions as well as two different containers. So it is useful for fixing an application [12].

Normally multi-tenancy is a single application which is accessed by many different users globally. In docker it is easy and inexpensive to create isolated environment for multiple instance of application tiers for each user. The processes are created and deployed in container itself making the booting up of operating system rapidly [12].

2. Research Method

A lot of research has been done regarding the study and analysis of docker and virtual machine. There are two groups of related works are described in this paper. First related work reports the updated performance comparison of virtual machine and Linux containers. Second related work narrates the research and implementation of cloud computing platform based on docker.

Wes Felter said that Isolation and resource control for cloud has been accomplished in the virtual machine. T.Swathi explains the virtualization in cloud computing [8]. For running an application in a virtual machine it gives extra level of abstraction so it will shrink the performance. So we are moving to containers [7].In this paper we analyse the requirement for cloud virtualization in both KVM and Linux containers. There has been more importance on the Linux containers than the KVM based on isolation and resource allocation. The result of KVM reveals that VM isolation adds overhead. It creates, deploys and updates the disk image which takes more time and space. So to overcome these problems we are traversing into Linux Containers. This blog [9] tells about the features of the containerization techniques and how to build and run the applications in the container. Hence the result of this work must be reconsidered exclusively to evaluate the virtual machine and Linux containers. The evaluations are based on CPU - Linpack, Memory bandwith-Stream, Random Memory Access - RandomAccess, Network bandwith - nuttcp, Network latency - netperf, Block io -fio, Redis and MySQL.

The work prepared closet to our study is that of Libin Zhao [10] .This study focuses on the docker application and advantages in cloud. The author studied about the operating system level virtualization on Linux. We studied about the features and components of docker in this blog [9]. Using Docker files to automate building of images reveals about the docker files and commands [10].In another blog we studied the 8-proven real-world ways to use docker [9].Here we focused on the difference between both virtual machine and docker, in this we found out docker is best for our research work. A docker offers a broad range of use cases so it is easy to use, access, and manage. We are analysing the comparison between docker and virtual machine in this [11]. Here we are also studying the process of deploying the images in container. This paper also describes how the images are pushed and parsed in Redis using dynamic library by the flex and bison [10]. Finally this paper completely explains about the application of docker in client especially in PaaS platform.

3 Proposed System

3.1 Core Technology

Docker is a tool which is used by Linux container. Linux containers are a virtualization method that is used for running multiple applications in a single host. This tool is discovered by the Dot Cloud. The basic components of docker are docker client, docker daemon, docker containers, docker images, and docker index and docker registry. Docker client acts as a link between user and docker daemon. Client gets the commands from the user and interacts with daemon bidirectionally. Docker daemon runs on the host machine. If any other component asks services to daemon, daemon acknowledges it. User cannot directly connect to the daemon, it has to go to the docker client and then to the user. Docker containers run components which run on any computer or on any infrastructure. It allows us to wrap all of its dependencies and libraries within single package. Each container is an isolated and secure application platform, so easy to run, move and delete. Docker images are build components used to create docker containers.It is read-only template which is easy to create or update images if required and can download other images created by user. Images are created automatically using file-housing instructions. Docker indeices are used for backing up docker images based on the registry access permissions. It distributes component which hold images. It serves huge collection of existing images for your use.

The efficient containerization features the docker uses are namespaces, control groups and union file system. To run an application docker provides various namespaces that helps to run the container in an isolated workspace. We can run many processes in a single container without affecting any other application process. Various namespaces are pid's purpose is for isolating one process to another, net is used for administrating network interfaces, ipc utilizes inter process communication resources, mnt is treated as managing mount points, uts(Unix Timesharing System) handles isolated kernel and identifiers[21]. The next important key feature is control groups (cgroups).Control group's duty is to share resources to the containers and confines the demanded resources. Another most efficient feature is Union File System (UFS).UFS constructs various layers so it is light weight and fast. UFS acts as building blocks for containers. The examples for UFS are AUFS (Advanced Union File System), BTRFS (B-tree File System), VFS (Virtual File System) and device mapper [21]. All these three features are integrated together in container format. The libcontainer is the default container format.

3.2 Docker Architecture

Docker configuration is based on the client server architecture basically docker client intimidates the docker daemon that creates the docker images which run inside the container. After the signal is received it has to create images which are captured by the daemon. To run an applications on a container, images are to be built first. Figure 1 explains how the client-server architecture works.

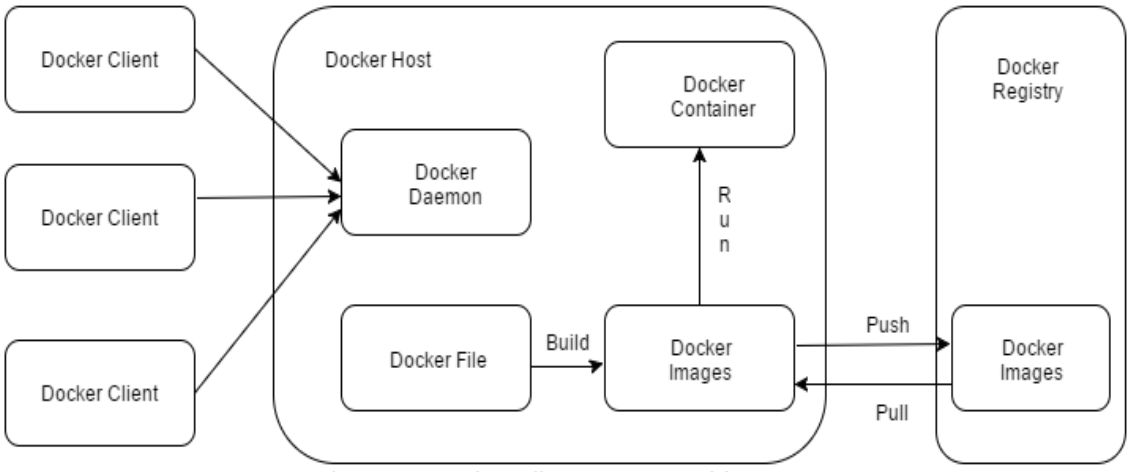

Figure 1: Docker client-server architecture

Docker images are read-only template that holds all related information needed to bootstrap a container. Each and every image starts from the base images and the read-only template is created by using the file-house instructions which are stored in docker file. For creating templates a new layer is created on that image .Once images are created it can be pushed into the registry.

Docker index allows backup of docker images which is available to others for use. It provides two-levels of access permissions one is public another one is private. To store images in private by using the private repositories which are only accessible by the members who have permissions. Public repositories are used for reusing and searching. To search for images in docker index, docker client can be used. If we launch the container a read-write layer is created on top of the original image. Then we can allocate a proper network and IP address that is needed for the application to run inside the containers. However we will communicate with other containers by using TCP. Those docker images can be shared via the docker hub or own registry.

3.3 Comparative study on docker and virtual machine

Comparing to traditional virtualization, docker is an enhanced technology. Figure 2 explains the architecture of both virtual machine and docker. In this hypervisor and guest operating system are replaced with docker engine. Docker engine takes account of all the application and its dependencies.

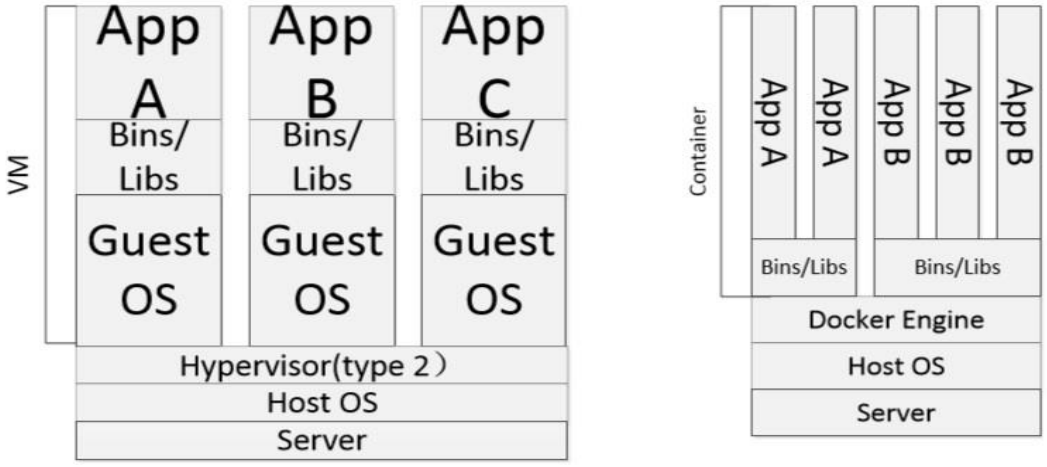

Figure 2: Virtual Machine Vs Containers

Virtual machine has a full operating system where all device drivers are installed in it. In VM resources are emulated to guest OS and hypervisor which makes it possible to run many instances on a single machine. Every guest OS runs as a single entity in a host system whereas docker containers are executed with the help of docker engine instead of hypervisor. Containers are smaller than virtual machines and they enable faster access with better performance and compatibility due to sharing of host. Hypervisor abstracts an entire device whereas containers just abstract the operating systems [17].Docker containers are much more potential than VM's. Docker containers are able to share single kernel and share application libraries. Containers present a low system overhead than VM. As the performance of an OS application in a container is better when same application running in VM. Containers are faster and resource oriented as users can stick to single platform for shared OS.A virtual machine can take several minutes to create and launch whereas containers can be created and launched in few seconds because in VM if we deploy images it might create duplicate images. Application running in containers offer superior performance compared to application running in VM.

3. Results and Analysis

In traditional system there were issues that were not convenient for users. This is because of the one to one relationship between physical computer and software. So the next generation was transformed to Virtualization. However virtualization helps to run more processes on the single machine. Virtual machines are heavy files and become inefficient to ship between hosts. It is also difficult to port from one place to another. To overcome these problems containerization plays the

major role. Containerization can afford Software as a service instead of Platform as a service. They are also lightweight and poratble.
In our work we are using CloudReports, a simulation graphic tool for cloud computing environment. It provides features like user-interface, report generation and API. So users can create their own extensions easily. CloudReport can be made easier to create and configure the simulation environment that can be deployed for later use. In this we found that containers are better than virtual machines in some cases. Our paper provides evaluations and tests which are based on CPU utilization, CPU latency, Image size, Booting up a container or virtual machine, Load average on host. Specifications are mentioned in below table [Table 1]

Table 1: System Specification

CPU		RAM Memory		OS Information	
Specification	Values	Specification	Values	Specification	Values
Architecture	X86_64	Total Memory #	264GB	Type	Windows
CPU#	40			Version	8.1
Threads per core	2			Docker client version	1.8.1
CPU GHz	2.40GHz			Docker server version	1.8.1
Model	Intel Core i5-4210U				

4. 1 CPU utilization
CPU Utilization is the significant constituent for performance. It is mainly used for assessing the behaviour of central processor unit's improvement on retrogression when running an application or piece of code. In docker CPU usage and memory usage are limited; therefore none of the process stays idle. Figure 3 describes the docker process was effectively utilized (theoretically 98.6%) by the CPU.

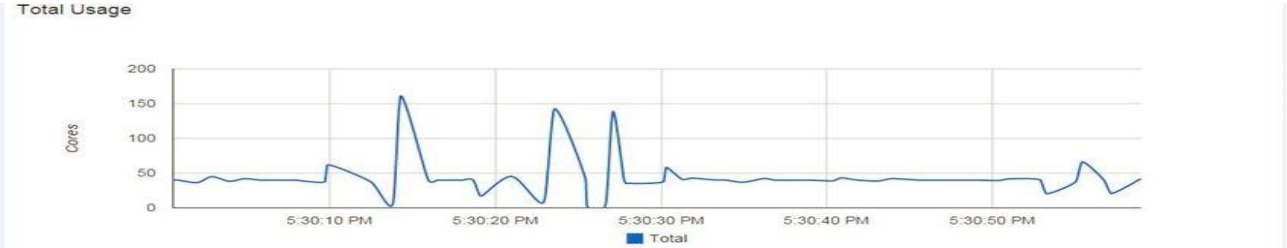

Figure 3: CPU Utilization

4.2 CPU Latency
CPU Latency is any delay or waiting that increases real or perceived response time beyond what is desired [14]. In Virtual machine CPU latency is higher than the docker. Latency issues are

hardware problem, heavy traffic and incorrect setup of configuration. So in Figure 4 CPU Latency is lower in docker compared to virtual machine.

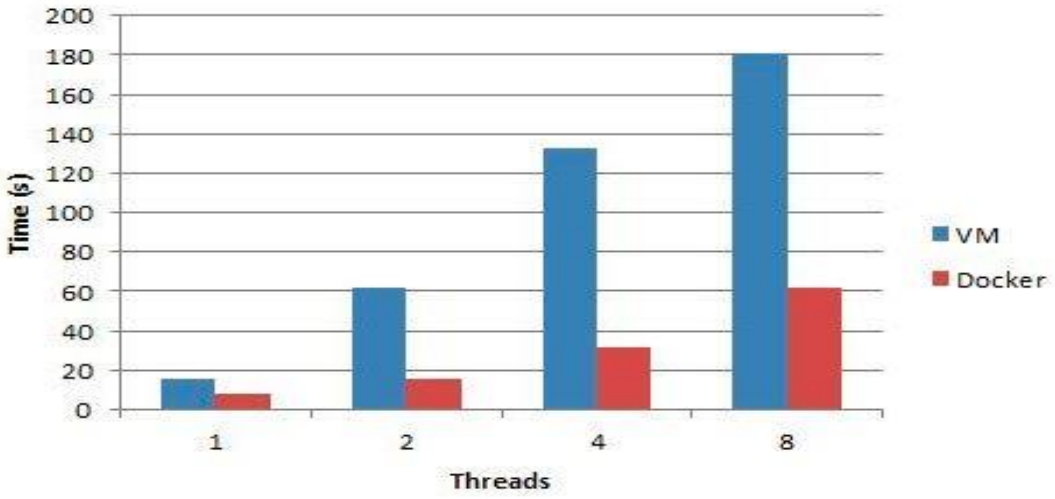

Figure 4: CPU Latency

4.3 Image Size

VM images are stored in .vmdk extension and it is heavier. In docker images, union file systems combine the serious of layers into single image. In spite of these layers docker images are light weight and it takes less storage space. So the docker images are 80% lighter than VM images. So in this figure [Figure 5] image size is measured.

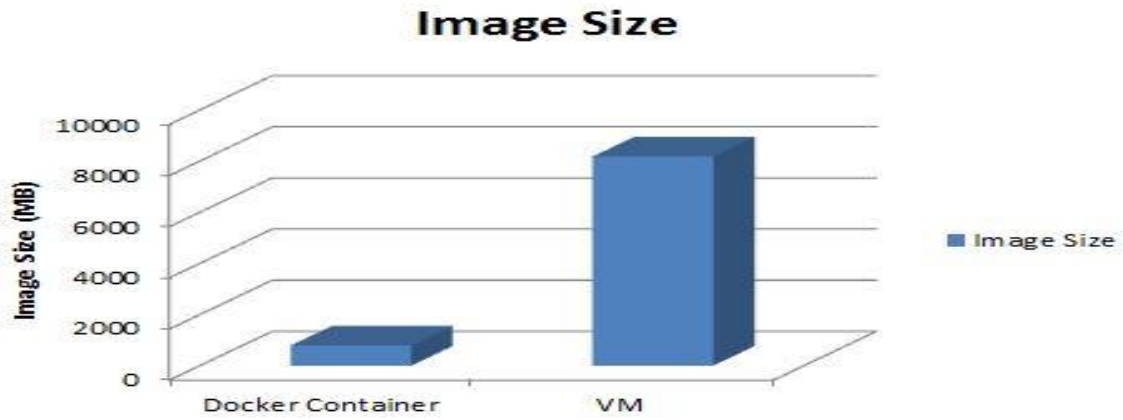

Figure 5: Image size

4.4 Booting up a container or VM

Docker isolates the application at file level. It feels like the real virtual machine. While creating the docker process, it makes the container but does not boot the operating system. So docker container takes only one by 20th of the time to start when compared to virtual machine. In this case time taken to boot up and shut down the containers increases exponentially as the number of containers increase [Figure 6].

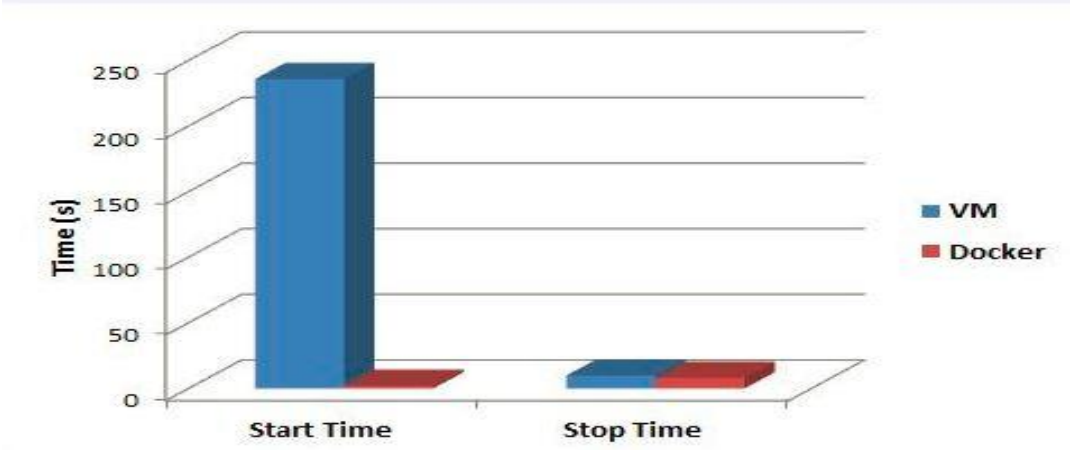

Figure 6: Booting up a container or VM

4.5 Load average on host

Load average is the value which represents the load on the system for a specific period of time [16]. When the number of processes is increases, size and complexity also rises. This will lead to overload. It has been reduced to half in docker comparing to virtual machine [Figure 7].

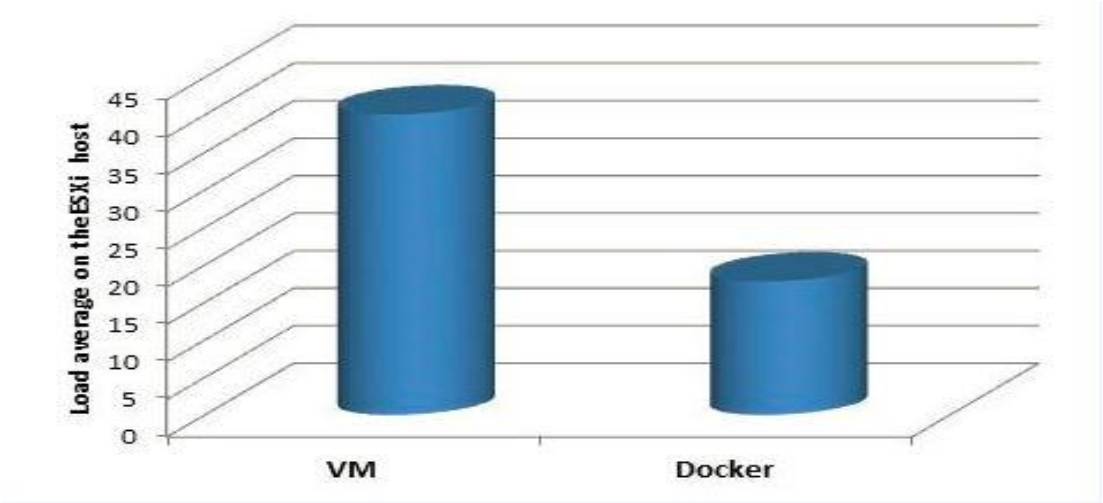

Figure 7: Load average on host

4. Conclusion

In Virtual machine there are some problems like performance degradation, lack of application portability, booting up time and memory usage. To overcome these we use a new tool called Docker which helps in enhancing the application performance. It is lightweight and portable, its performance is better when compared to virtual machine. In our project we used CloudReport Simulator to compare the performance between both docker and virtual machine. Based on our results we found that docker certainly has an advantage when compared to virtual machine.

6 Future Work

The future work on docker is regarding the storage of information inside the containers. There are two ways to store the data processed by the applications

1. Data can be stored externally in data volumes and the volumes can be mounted to the container using mount points. As the storage is external the data is not portable along with the applications. It should be shipped separately.

2. Data volumes can be included inside the Docker File and when the file is deployed that include(s) a data volume in which the data can be stored. In this type the data along with the applications is portable.

Networking of Docker also needs to be focused for future work. Docker containers are not transparent to networks in which they are in. Docker uses the host IP address and its NIC card to communicate with other hosts in the network. As the assigned IP will change as the host moves from one host to other resulting in a communication problem.

References

[1] Sangeetha K.S, Prakash P, “Big Data and Cloud: A survey”, Advantages in Intelligent Systems and Computing (AISC), Vol. 325, pp. 773-778, Springer India 2015.
[2] Kumar, A., Sathasivam, C., Periyasamy, P. “Virtual machine placement in cloud computing (2016) Indian Journal of Science and Technology, 9 (29), art. no. 79768.
[3] Is the Cloud the future of computing Joseph M.Kizza University of Tennessee-Chattanooga,USA Li Yang University of Tennessee-Chattanooga,USA
[4] A Study on Virtualization Techniques and Challenges In Cloud Computing Durairaj. M, Kannan.P
[5] http://www.losangelescomputerhelp.com/2010/04/the-different-types-of-virtualization/
[6] http://www.iscsi.com/resources/Storage-Virtualization.asp
[7] https://docs.docker.com/
[8] Virtualization in cloud computing T.Swathi, K.Srikanth, S. Raghunath Reddy
[9] http://blog.flux7.com/blogs/docker/docker-tutorial-series-part-1-an-introduction
[10]https://www.digitalocean.com/community/tutorials/docker-explained-using-dockerfiles-to-automate-building-of-images
[11]https://access.redhat.com/documentation/enUS/Red_Hat_Enterprise_Linux/7/html/7.0_Release_Notes/sect-Red_Hat_Enterprise_Linux-7.0_Release_Notes-Linux_Containers_with_Docker_Format-Advantages_of_Using_Docker.html
[12] http://blog.flux7.com/blogs/docker/8-ways-to-use-docker-in-the-real-world
[13]http://blogs.technet.com/b/winserverperformance/archive/2009/08/06/interpreting-cpu-utilization-for-performance-analysis.aspx
[14]https://www.safaribooksonline.com/library/view/cassandra-high performance/9781849515122/ch12s03.html
[15] http://whatis.techtarget.com/definition/latency
[16] http://www.golinuxhub.com/2014/06/understanding-load-average-in-linux-and.html
[17] http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/
[18] Diliu , Libin Zhao, ”The Research and Implementation Of Cloud Computing Platform Based On Docker”
[19] Wes Felter,Alexandre Ferreira, Ram Rajamony, Juan Rubio , ”An Updated Performance Comparison of Virtual Machines and Linux Containers”
[20] Robert P Goldberg, "Survey on Virtual machine Research"
[21] https://docs.docker.com/engine/introduction/understanding-docker/
[22] http://link.springer.com/chapter/10.1007%2F978-3-319-10530-7_6#page-1