Containers & Virtual machines

A performance, resource & power consumption comparison

Martin Lindström

June, 2022

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the bachelor's degree in software engineering. The thesis is equivalent to 10 weeks of full-time studies.

Contact information:

Author:

Martin Lindström

Email: mamv18@student.bth.se

University advisor:

Andreas Arnesson

Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 5

Abstract

Due to the growth of cloud computing in recent years, the use of virtualization has exploded. Virtual machines (VMs) and containers are both virtualization technologies used to create isolated computing environments. While VMs are created and managed by hypervisors and need their own full guest operating system, containers share the kernel of the host computers and do not need a full guest operating system. Because of this, containers are rumored to have less overhead involved, yielding higher performance and less resource usage compared to VMs.

In this paper we perform a literature study along with an empirical study to examine the differences between containers and virtual machines when it comes to cpu, memory and disk **performance**, cpu and memory **resource utilization**, and **power consumption**. To answer the question regarding performance, a series of benchmarks were run inside both a container and a VM. During these benchmarks the resource utilization of the host machine was also measured to answer the second question and to answer the third and final question the power draw was measured while some of the benchmarks were running.

The results showed that the cpu performance was extremely similar between the two and memory performance seemed to be similar for the most part but fairly big differences were seen in favor of both depending on the benchmark in some cases. With disk performance the container was between 15-50% faster depending on the benchmark. As for resource usage, the cpu usage was the same for both technologies but memory usage differed greatly in favor of the container. The VM used between 3-4 GiB and the container between 70 MiB - 2.5 GiB depending on the benchmark. The power draw was the same for both technologies when under cpu and memory load but when idle the VM proved to draw around 40% more power.

Keywords: Containers, Virtual Machines, Virtualization, Hypervisors, Performance

Contents

Chapter 1

Introduction

1.1 Background

Virtual machines are used extensively in cloud computing, and are especially synonymous with Infrastructure as a Service (IaaS) [27, 28]. Cloud platforms such as Microsoft Azure make VMs available to their customers and many Platform as a Service (PaaS) and Software as a Service (SaaS) providers are built on IaaS with all their workloads running inside VMs [28]. In the last decade or so, container based virtualization has become popular, offering an alternative to hypervisor based virtualization. With the rise of containers, application development and deployment has been greatly improved [1, 29]. Containers have incredibly short boot time, are easy to tear down and replace and are highly portable which has led to an incredible adoption rate of containerization technology, with Docker being the most popular [15]. Due to the ever expanding need for IT-services, applications and platforms, we are determined to compare the performance, resource usage and energy consumption differences between these two virtualization technologies to see if one might be preferred over in those areas.

1.1.1 Virtual machines

The history of the virtual machine dates back to the 1960's and in the 1990's the modern take on it, the one we use today, was introduced [8]. A virtual machine (VM) is a virtual environment that functions as a virtual computer with its own CPU, memory, network interface and storage [2]. It uses software instead of hardware to run programs and is most commonly used to isolate applications from the host machine or for server virtualization [2, 26]. Virtual machines make it possible to run several different "computers", each isolated from the host with their own operating system, on top of a single physical computer. A virtual machine is created and managed by what's called a hypervisor [3], which is a software that allows the host computer to virtually share its resources, such as memory and processing, with multiple guest VMs. Figure 1 displays an example of the basic architecture of hypervisor-based virtual machines.

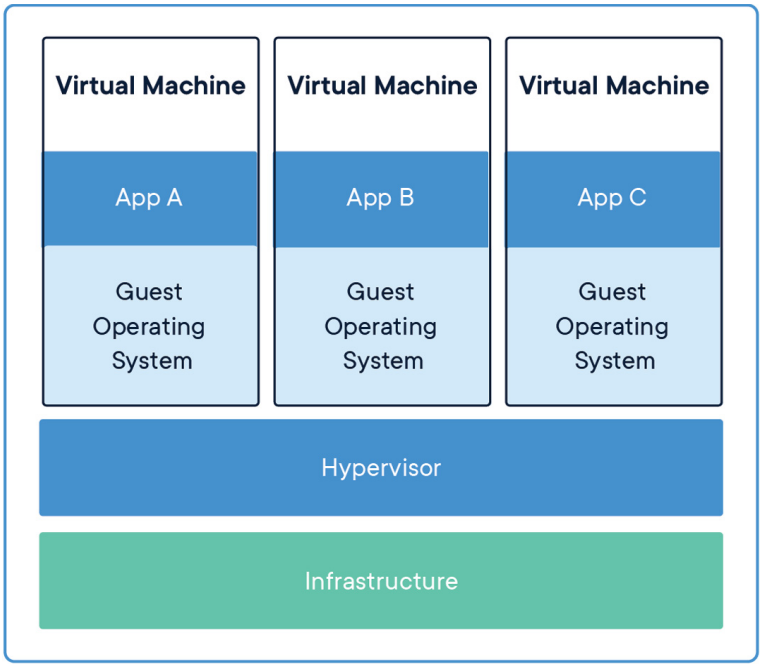

Figure 1 - Virtual machine architecture example

1.1.2 Containers

Containers as we know them first saw the light of day in 2008 with Linux containers (LXC) but it was the launch of Docker in 2013 that really made the adoption of containerization technology explode [1, 4]. Containers are packaged units of software that contain the application code along with all of the dependencies it needs to be able to run. Just like virtual machines, containers are isolated from the other processes on the host machine but unlike virtual machines they don't need an entire guest OS to run, making them more lightweight and portable [1, 12]. As seen in figure 2, multiple containers can run on a single operating system without the need for their own.

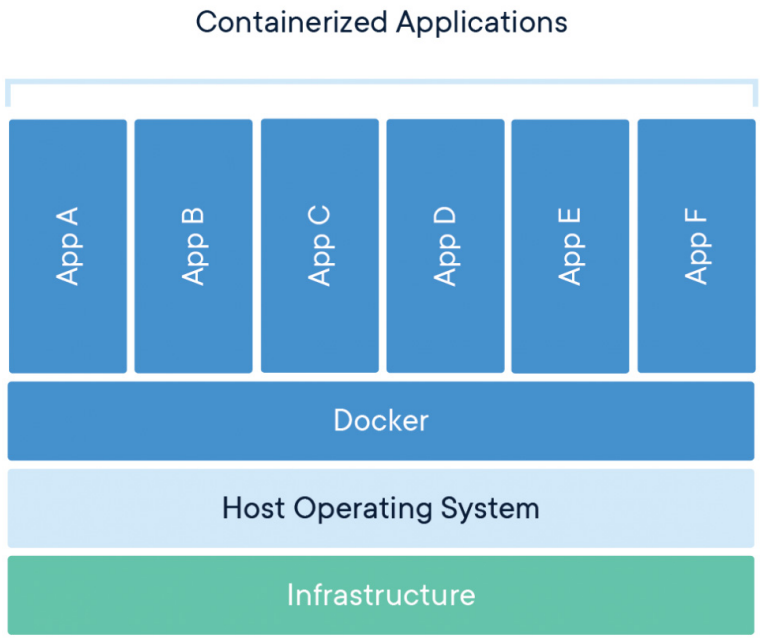

Figure 2 - Docker container architecture example

1.2 Purpose

Because VMs and containers are two popular virtualization technologies we wanted to compare certain aspects of them in order to get an increased understanding of them. VMs for example have a reputation of introducing more overhead [30, 31] which is something that we want to investigate to see how this impacts the resource usage and performance compared to containers that do not introduce the same overhead. Another aspect of this study is power consumption. With an increasing need for computing power in our society, data centers grow bigger and bigger to accommodate this growth. Finding ways to make these data centers more efficient is an important task, both in terms of economical as well as environmental purposes.

1.3 Scope

This study will not focus on whether containers are better than virtual machines or vice versa for specific use cases. The fact that containers are faster to boot up and easier to deploy or that virtual machines are more secure for example is generally known and each of them have their use cases [29]. The scope of this study is strictly limited to examining the differences between the two technologies when it comes to cpu, memory and disk performance, cpu and memory resource usage and power consumption.

Chapter 2

Research Questions

Both virtual machines and containers are heavily used virtualization technologies so examining how they perform in different regards could be valuable in terms of future improvements and iterations of the technologies. There are three different questions that this study aims to answer regarding performance, resource usage and power draw.

RQ1: How does the CPU, memory and disk performance differ between containers and virtual machines?

The study will examine how containers and virtual machines differ when it comes to cpu, memory and disk performance. Containers have a reputation to have faster performance compared to VMs due to having less overhead which is something that we want to determine if it holds true or not.

Performance in this case can either mean amount of operations / timeframe or time taken to complete a certain task.

RQ2: How does the CPU and memory resource usage differ between containers and virtual machines?

This study will also examine how the amount of resources used by the VM and the container differ. Because VMs need an entire guest operating system and containers do not, we want to look at how this affects the amount of resources being used.

Resource usage refers to how much of the host machine's cpu or memory is being utilized.

RQ3: How does the power consumption differ between containers and virtual machines?

Lastly, With the importance of energy efficiency, this study intends to examine the differences in how much power each technology consumes. Answering this question could be important for both cost effectiveness and environmental purposes.

Chapter 3

Research Method

3.1 Literature Study

A literature study was conducted to see what previous studies and research concluded regarding performance, resource utilization and power consumption of virtual machines and containers. The method for the literature study consisted of searching for research papers and articles in databases such as BTH Summon, Scopus and IEEE Xplore along with Google search engine. The searches consisted of the following words or a combination of the following words: “Containers”, “Virtual machines”, “Performance”, “Power consumption”, “Resource utilization”, “Containers vs Virtual machines”, “Docker”. When finding relevant research papers, the snowball method was used to find additional studies and research papers.

3.2 Experiments

A series of experiments was conducted to answer the three research questions of this study. The experiments were divided into three parts, **performance, resource utilization** and **power consumption**. To carry out the experiments two different environments were set up on the host computer, one Docker container and one VirtualBox [19] virtual machine (With GUI), both running Ubuntu 20.04. Both environments were configured with 4 cpus and 4GB of system memory (RAM).

Host computer specifications:

- **OS:** Ubuntu 20.04
- **CPU:** Intel i5 6500T @ 2.5Ghz, 4 cores/8 threads
- **Memory:** 16GB DDR4 2133Mhz
- **Storage disk:** 128GB SSD

3.2.1 Performance

To answer the first research question “How does the CPU, memory and disk performance differ between containers and virtual machines?”, a series of benchmark tests from phoronix-test-suite [15] were installed and run in both the Docker container and in the virtual machine. Three different CPU benchmarks, three different memory benchmarks and two different disk benchmark tests were used to get a bigger and more varied array of data. 7-zip, N-Queens and RAMspeed benchmark tests were chosen because they were used in some of the other studies with the reasoning that we wanted to see if we got similar results to them or not. The other tests were chosen to provide something “new”. Different CPU, memory and disk tests might utilize the

computer in different ways and show different results which is why this decision was made. More information about the tests can be found on openbenchmarking [23].

After booting up the container or VM there was a waiting period of a couple of minutes before starting the tests so that the machines could settle down first. Each benchmark test was automatically run a minimum of three times, sometimes more depending on if the results of the tests deviated too much. The container/VM was not restarted after every test but there was a waiting period between tests to let the machines settle down.

The following benchmarks were used:

CPU:

- 7-zip compression
    - 7-Zip is a benchmark that measures the computer's ability to compress and decompress files. It measures this ability in MIPS, million instructions per second. Data compression is the process of encoding information using fewer bits than the original representation in order to reduce the file size [25].
- N-Queens
    - N-Queens is a benchmark that measures the amount of time it takes to solve the N-Queens puzzle. N-Queens is the problem of placing N queens on an N x N chessboard in such a manner that no queens can attack each other.
- Kvazaar
    - Video encoding is the process of converting digital video files from one video format to another and Kvazaar is a video encoding benchmark that measures the encoding performance by the amount of fps (frames per second) that the computer is capable of outputting.
    - Preset: medium
    - One test with 1080p, one test with 4K

Memory:

- MBW
    - MBW measures the memory "copy" bandwidth performance by copying files from one memory location to another.
    - Fixed block size, one test with 128MB array size, one test with 1024MB array size
- RAMspeed
    - RAMspeed tests the systems memory (RAM) performance

    - One test with **Add** (which reads data from the first memory location, then reads from the second location and finally adds them up and writes the result to the third location (A = B + C)) and one test with **Copy** (which transfers data from one memory location to another, i.e copies it (A = B)), both using integers.

- Tinymembench
    - Tinymembench measures the memory “copy” performance by copying files from one memory location to another. Also uses Memset (Which functionality is not explained by the benchmark) to measure performance.
    - One test with Memset, one with Memcpy

Disk:

- PostMark
    - PostMark is a benchmark designed to simulate small-file testing similar to the tasks endured by web and mail servers. The test profile will set PostMark to perform 25,000 transactions with 500 files simultaneously with the file sizes ranging between 5 and 512 kilobytes.
- Dbench
    - Dbench is a tool to generate I/O (input/output) workloads to the filesystem. At the heart of DBENCH lies the concept of a "loadfile". A loadfile is a sequence of operations to be performed one statement at a time. This could be operations such as "Open file XYZ", "Read 5 bytes from offset ABC", "Close the file" etc.
    - Three separate tests with 6, 48 and 256 clients

3.2.2 Resource Utilization

To answer the second research question “How does the CPU and memory usage differ between containers and virtual machines?”, the host machine’s cpu and memory usage was measured while the performance benchmarks from the first experiment were running in a Docker container as well as when they were running in a virtual machine. These measurements were done manually and the methods differed for each environment. To measure the cpu and memory usage of the virtual machine, Ubuntu’s system monitor was used. This tool didn’t show the resources used by Docker though, therefore Docker’s command “Docker stats” was used to measure the resources used by the container.

Additionally, a simple web application was built with Express.js and SQLite which displays some data from the database. This application was then run in both

environments while being under load with simulated traffic using Apache Benchmark [21] and the host machine's cpu and memory resources were measured during this time. The application was subjected to 10, 100 and 1000 concurrent connections for 60 seconds each in three separate tests.

3.2.3 Power Consumption

To answer the third and final research question "How does the power consumption differ between containers and virtual machines?", a tool called powerstat [20] was used to measure the power consumption of the host machine while running either the Docker container or the virtual machine. The power consumption of the host machine was measured for 120 seconds while A) no applications were running, B) Docker container and virtual machine were running under no load, separately, C) The Docker container and virtual machine were put under load by running the web application with 100 concurrent users, 7-Zip, MBW and PostMark benchmarks, separately.

Chapter 4

Literature Review

Searching for papers that were relevant to the study revealed that there are quite a few previous studies done on both the power consumption of containers and virtual machines as well as the performance differences between them. While some of these also bring up the difference in resource usage, it proved more difficult to find studies dedicated to it. Not all studies that we found were relevant to us however. When looking at performance for example, we only chose the ones that included cpu, memory or disk performance. Studies that were only looking at network performance were disregarded. In total we found twelve studies that we deemed relevant and this chapter will summarize the findings of these studies.

According to a paper by A. M. Potdar et al [5] in which they compared the CPU, memory and storage read/write performance of Docker containers and virtual machines by running different benchmarks on the systems, Docker containers proved to have the better performance in all categories. One of the benchmarks that tested the CPU calculated prime numbers up to a maximum of 50,000 and showed that the Docker container took roughly 19 seconds to finish the benchmark while the VM took about 37 seconds to finish, making the Docker container almost twice as fast in that particular test. Another CPU benchmark they ran was 7-zip compression in which the container performed ~8200 MIPS (million instructions per second) and the VM performed only 2000 MIPS, making it four times slower than the container. The memory test which measured RAM speed (MB/second) resulted in the container being almost three times faster than the VM and in the disk benchmark the container was about twice as fast.

In a paper by S. Shirinbab et al [9], the authors measured the performance differences (transactions per second, TPS) when running Cassandra databases in three different environments, a native non-virtualized environment, Docker containers and virtual machines by writing to and reading from the databases. They also measured the CPU utilization while doing this which showed that the VM's operation and resource overhead severely limits the performance compared to containers. One test showed that while performing 40.000 TPS the native machine utilized ~35% of the CPU while the Docker container utilized ~42% of the CPU and the VM utilized ~60% of the CPU. A. M. Joy came to a similar conclusion regarding the performance differences in her study [11]. In her experiment she set up a frontend application along with a backend server which hosted a database for storing and retrieving data for the frontend

application. The application was then stress tested by sending simultaneous requests to it. The results showed that the Docker container was able to process 10.000 requests within 600 seconds while the VM was only able to process ~1600 requests in the same timeframe.

Preeth E N et al [13] found similar results to S.Shirinbab et al regarding container vs native performance when evaluating Docker containers based on hardware utilization. They benchmarked the containers with Bonnie++, a benchmarking suite which aims to test the hard drive and file system performance. In their tests, they used a fixed data size of 40MB and they then measured the read and write performance of the container and compared it to the host machine's native performance. They came to the conclusion the Docker container came close to native performance.

Contrary to A. M. Potdar et al [5] and S. Shirinbab et al [9] which both showed that virtual machines introduce significant overhead when it comes to CPU and memory performance, W. Felter et al [10] came to the conclusion that their virtual machine only introduced negligible overhead when it came to CPU and memory usage and performance. This was however after an extensive and careful configuration of the VM. Despite that, Docker containers proved to perform as well or better than the virtual machine in all of their tests, which is in line with the other papers.

R. Morabito et al compared performance differences between a native machine, Docker container, LXC container and a virtual machine in their paper [16]. They focused on measuring the CPU, memory, disk I/O and network I/O performance for each technology by running a series of benchmarks. One of the tests they used for the CPU was a multi-threaded tool which calculates the value of Pi. This test showed that the native system was the fastest, then came the containers and lastly the vm. The differences in cpu and memory performance were not substantial but in the case for disk performance the virtual machine was between 35-50% slower.

When it comes to power draw, G. Anusooya & V. Vijayakumar's results showed that virtual machines draw a lot more power compared to containers, at least when it comes to network load [6]. They tested this by measuring the power draw of the host machine while running a certain amount of virtual machines and containers, with each vm/container running one application each. The results showed that running one virtual machine with one application drew more power than four containers, each running one application, combined. This result however is not at all in line with Congfeng Jian et als results [7]. It should be noted though, that Congfeng Jian et al measured power draw during cpu and memory intensive operations and not network intensive. In their experiments they compared the power consumption between four different hypervisors, KVM, ESXi, Hyper-V, XenServer along with Docker on

different platforms such as rackservers, a desktop and a laptop. To stress the systems two different workloads were introduced, one cpu heavy workload which calculated prime numbers and one web server with a database where a script inserted 10.000 rows of data. Four different levels of workloads where each level introduced more virtual machines/containers were used when testing on each platform. The results showed that no specific hypervisor used less or more power than any of the others. The hypervisor/container that consumed the least amount of power differed for each of the different levels of workloads as well as for each different platform they were being run on. In some cases the Docker container consumed less power than the virtual machines and in some cases some of the virtual machines consumed less power than the Docker container.

R. Morabito et al [17] found similar results in their paper regarding the power consumption. In their experiments they compared the power consumption of 4 different virtualization technologies, two hypervisors (KVM and Xen) and two container technologies (Docker and LXC) while running one CPU benchmark, one memory benchmark and one network benchmark. These benchmarks were run in four different test cases, one with only one guest running, one with two guests running, one with four guests running and one with eight guests running simultaneously. According to the results the difference in power consumption was negligible between all of them in the CPU and memory benchmarks but in the network benchmarks the virtual machines consumed more power than the containers. This was accredited to the fact that the network packets have to be processed by extra layers in the virtual machines compared to the containers.

S. S. Tadasse et al [18] experimented with comparing the resource usage and power consumption of virtualbox VM’s and Docker containers and their results showed that their virtual machines consumed more power than their Docker containers while idle. When increasing the amount of virtual machines running, the CPU usage of the host machine along with the power consumption also increased, even if the virtual machines were idle. This was not the case for the containers though. Increasing the amount of containers running did not increase the CPU usage and therefore not the power consumption. However when putting the virtual machines and containers under CPU load they consumed the same amount of CPU resources and power, and when putting them under network load the virtual machines consumed more power than the containers, which is similar to the results in most of the other studies.

Chapter 5

Results

5.1 RQ1: How does the CPU, memory and disk performance differ between containers and virtual machines?

5.1.1 CPU performance

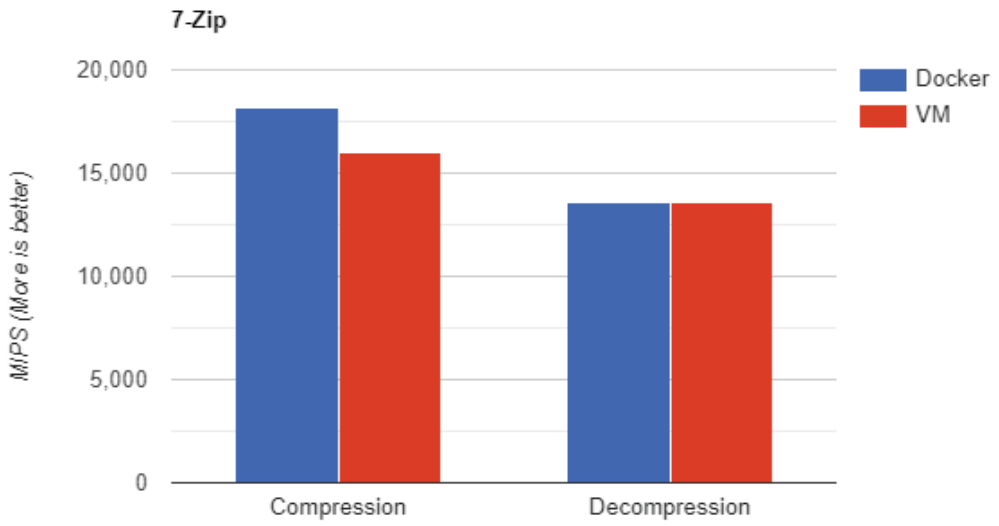

Figure 3 - 7-Zip compression and decompression cpu performance benchmark results

Figure 3 shows that when it comes to compression, the container is roughly ~12% faster than the VM but when it comes to decompression they are virtually identical.

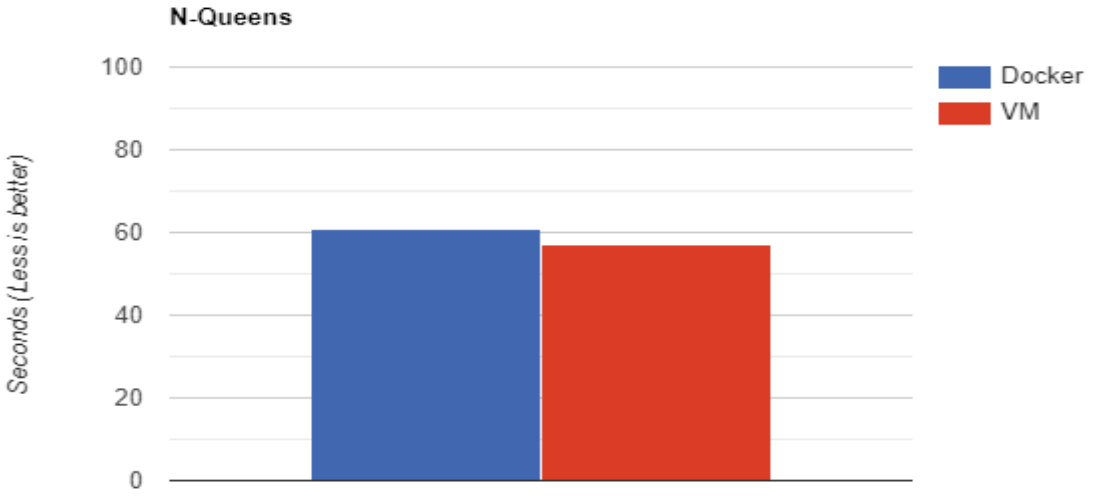

Figure 4 - N-Queens cpu performance benchmark results

In figure 4 it can be seen that the VM actually solved this problem a bit faster than the container, however the difference is only roughly around ~5%.

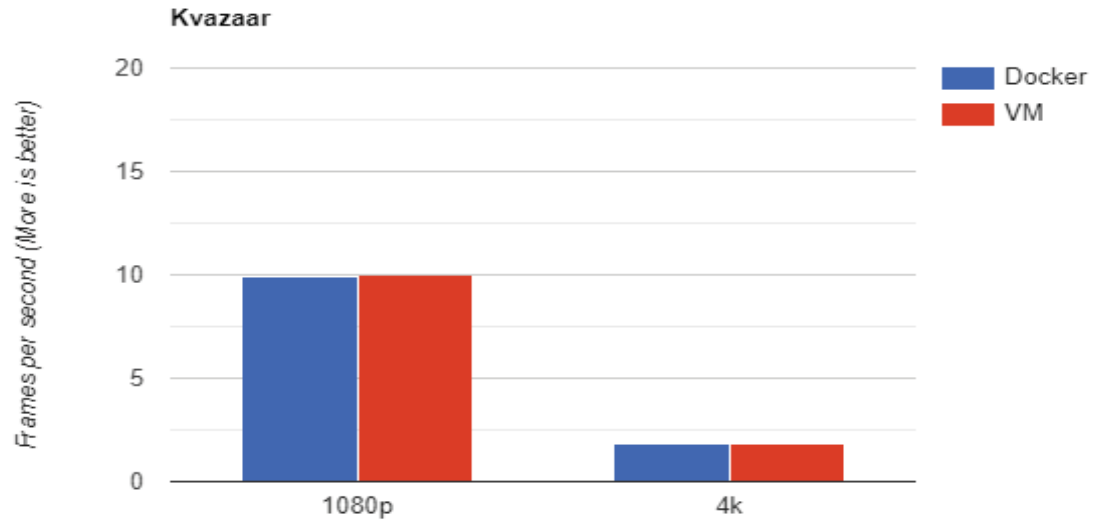

Figure 5 - Kvazaar cpu performance benchmark results

As seen in figure 5 the performance of the container and VM is almost identical with a negligible difference in the 1080p test.

5.1.2 Memory performance

Figure 6 - MBW memory performance benchmark results

In figure 6 we can see that the VM is outperforming the container by a fairly big margin which is quite surprising. The reason for this is unknown and it isn't in line with the other memory benchmarks.

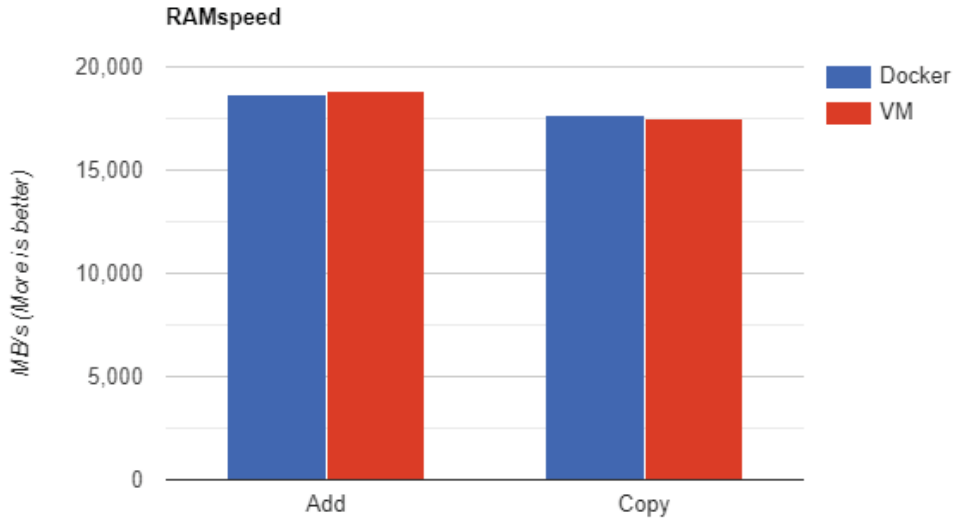

Figure 7 - RAMspeed memory performance benchmark results

The results seen in figure 7 show that the performance is almost identical for both tests. It's interesting to see that the results of two different benchmarks, MBW and RAMspeed, differ that much even though both perform memory copy operations.

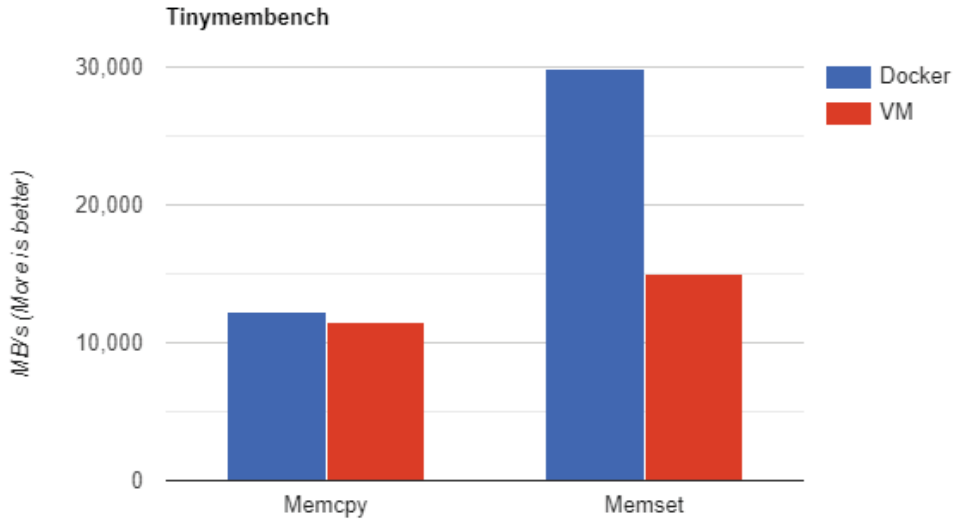

Figure 8 - Tinymembench memory performance benchmark results

In figure 8 we can see that the container beat the VM by a small margin in the “copy” test but in the “Memset” test the docker container performed around ~50% better.

5.1.3 Disk performance

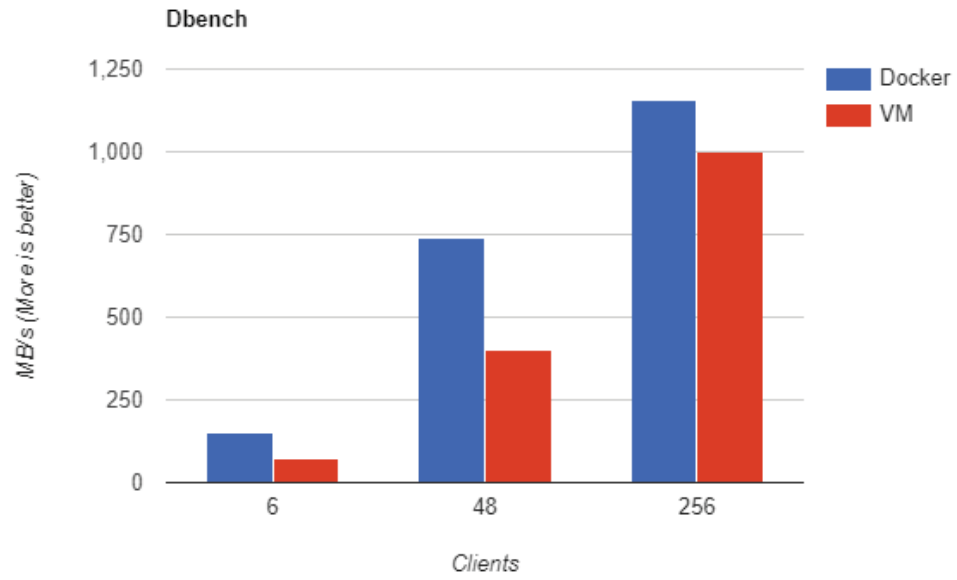

Figure 9 - Dbench disk performance benchmark result

As seen in figure 9, this benchmark was run three separate times with varying amounts of clients and the docker container performed best in all three tests.

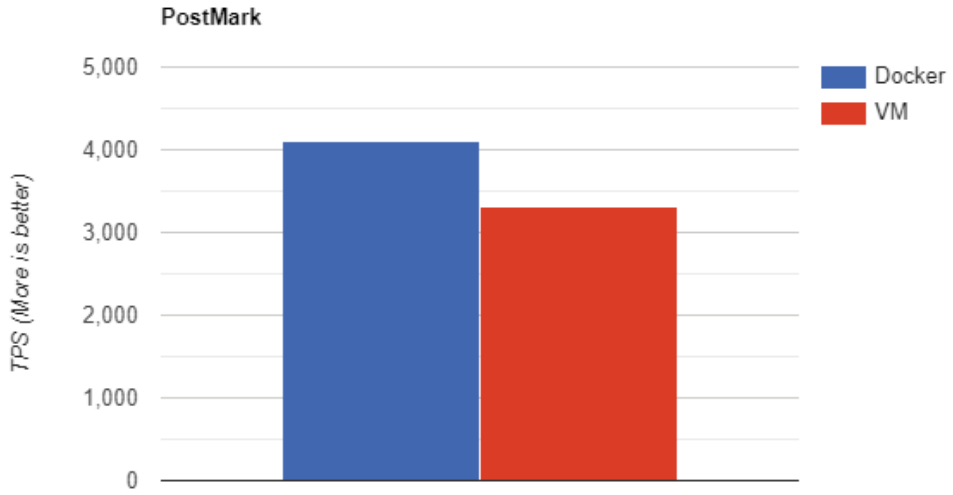

Figure 10 - PostMark disk performance results

Figure 10 shows that the docker container performs roughly ~30% better in that particular test.

5.2 RQ2: How does the CPU and memory resource usage differ between containers and virtual machines?

5.2.1 CPU usage

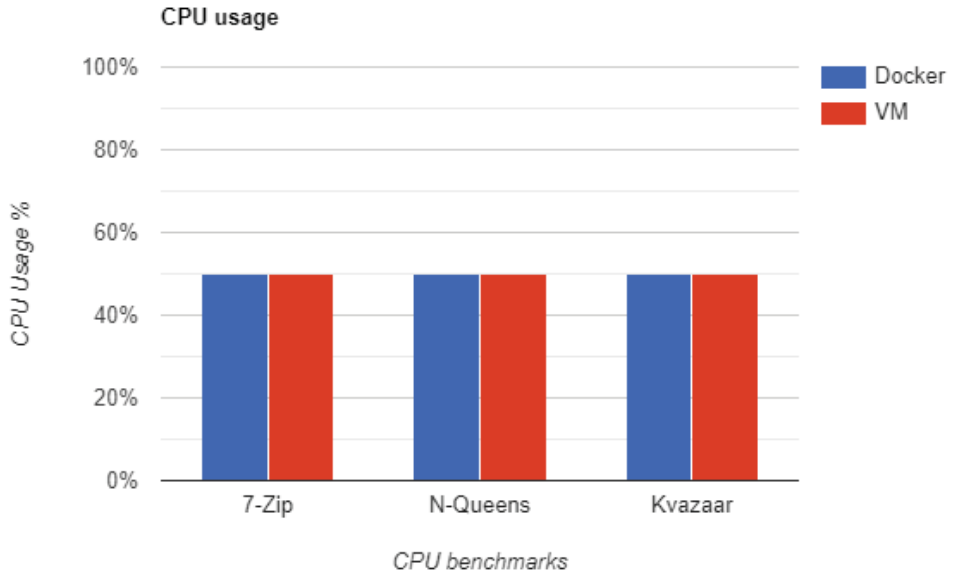

Figure 11 - CPU usage during CPU benchmarks

As seen in figure 11, the cpu usage of the host machine was exactly the same while running the docker container and the virtual machine. This makes sense considering that the cpu benchmarks are using 100% of the available cpu resources and that both the container and the vm were given 4 cpus, which is half of the available cpus of the host machine, i. e. 50%.

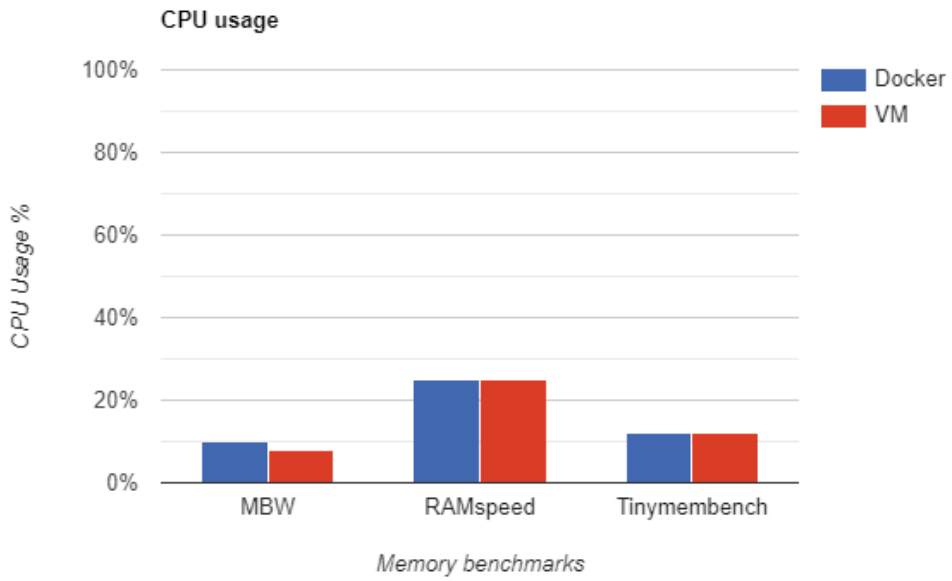

Figure 12 - CPU usage during memory benchmarks

In figure 12 we see again that the cpu usage is about the same for the container and the virtual machine, with a slight difference of about ~2% in the MBW benchmark.

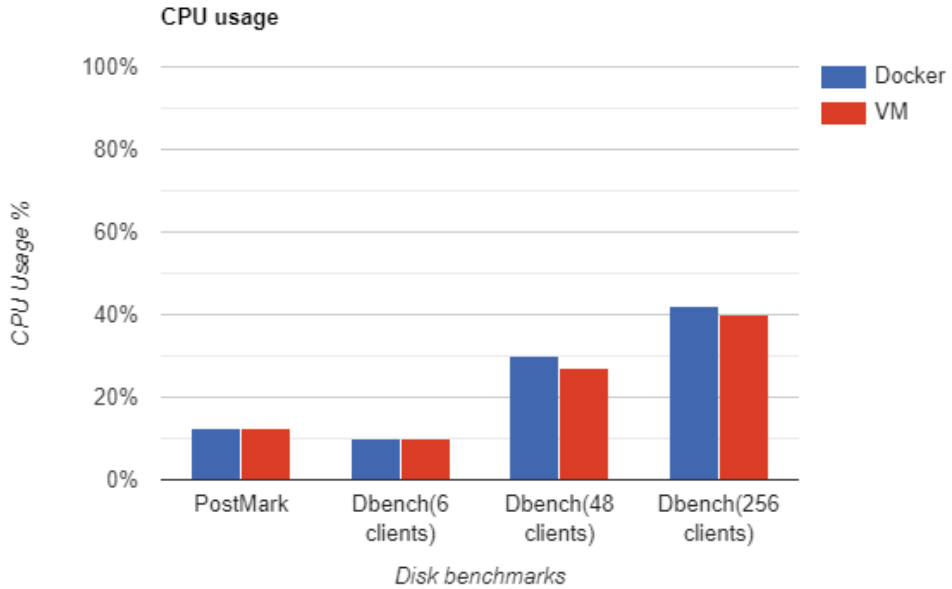

Figure 13 - CPU usage during disk benchmarks

A similar result can be seen in figure 13, which shows the results while running the disk benchmarks. Surprisingly though, the docker container used around 3-5% more cpu resources than the vm in two of the tests. However, this could potentially (but not necessarily) be due to the fact that the readings were done manually and that the cpu usage fluctuated throughout the tests making it difficult to get a 100% accurate reading.

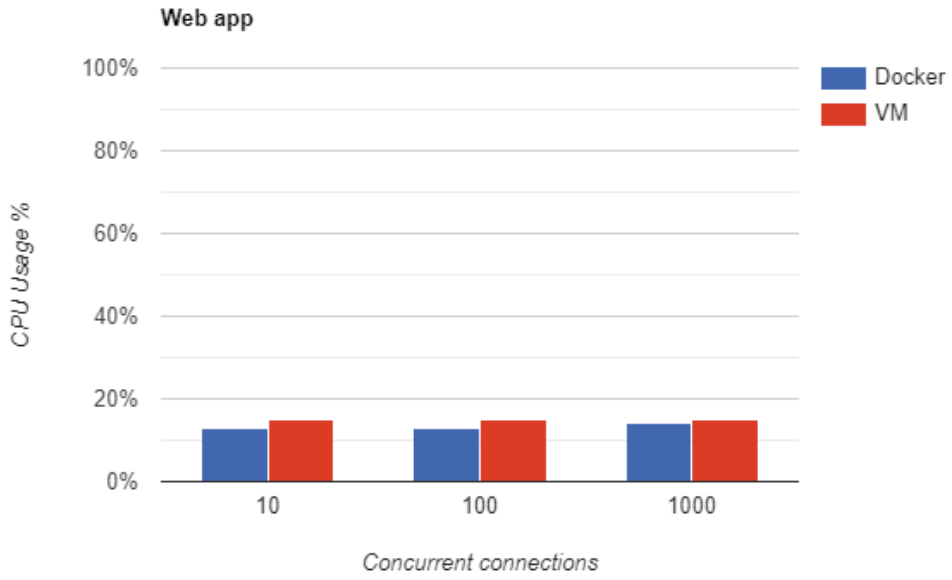

Figure 14 - CPU usage while simulating traffic to the web app

Figure 14 shows the cpu usage while running the created web application with 10, 100 and 1000 concurrent users. These results have the vm as the one using more cpu resources but this difference was only roughly around 1-3%, which is fairly insignificant.

5.2.2 Memory usage

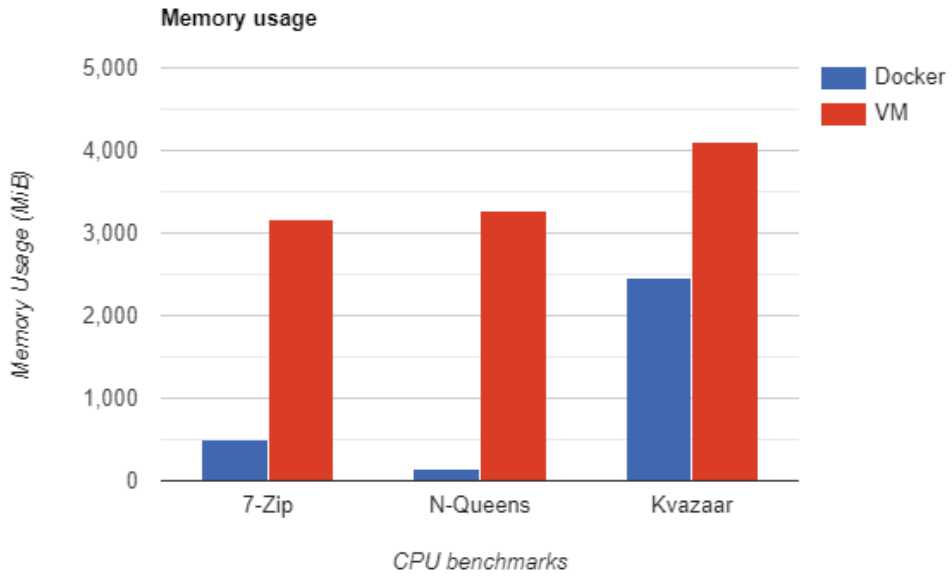

Figure 15 - Memory usage during CPU benchmarks

The memory usage tells a very different story compared to cpu usage. Figure 15 shows that the memory usage is significantly higher for the vm while running the cpu benchmarks. The virtual machine used between 3.1GiB - 4.1GiB while the container only used between 100MiB - 2.4GiB depending on the benchmark.

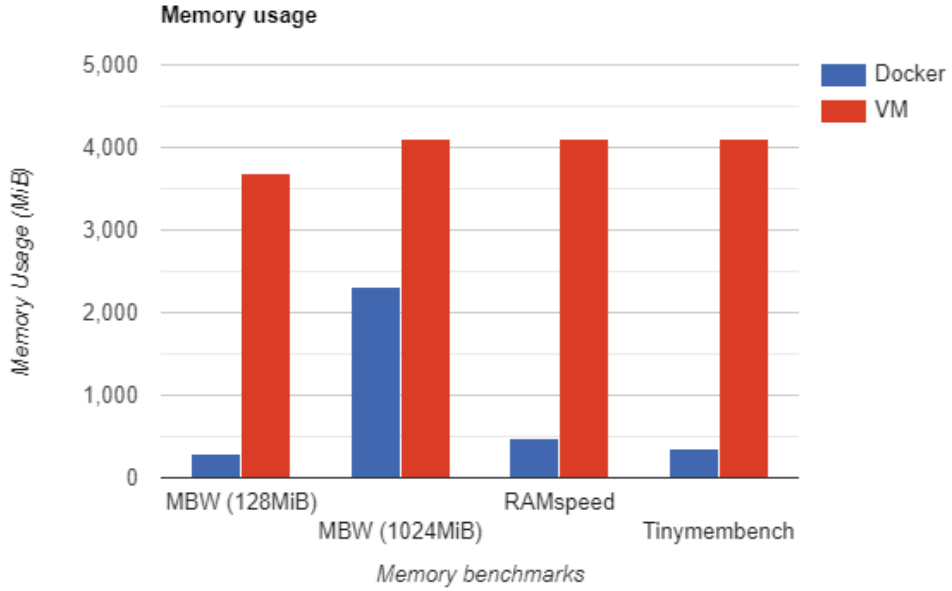

Figure 16 - Memory usage during memory benchmarks

The same results can be seen in figure 16. The container uses around ~250MiB - 500MiB in three of the tests and 2.2GiB in one while the virtual machine is hovering around that 4GiB mark in all of the memory tests.

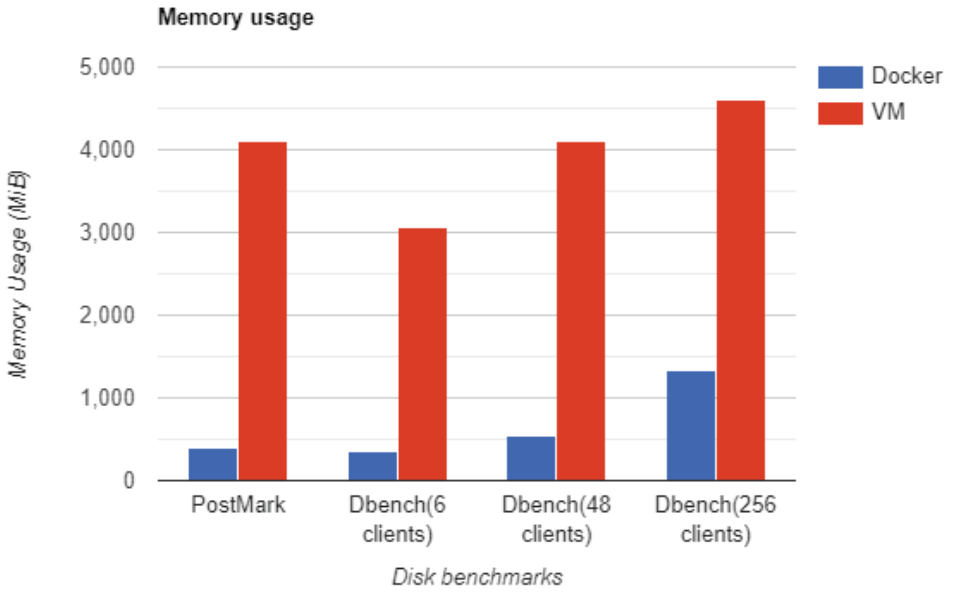

Figure 17 - Memory usage during disk benchmarks

Again, the same results present themselves while running the disk benchmarks in figure 17 as well as when running the web app seen in figure 18.

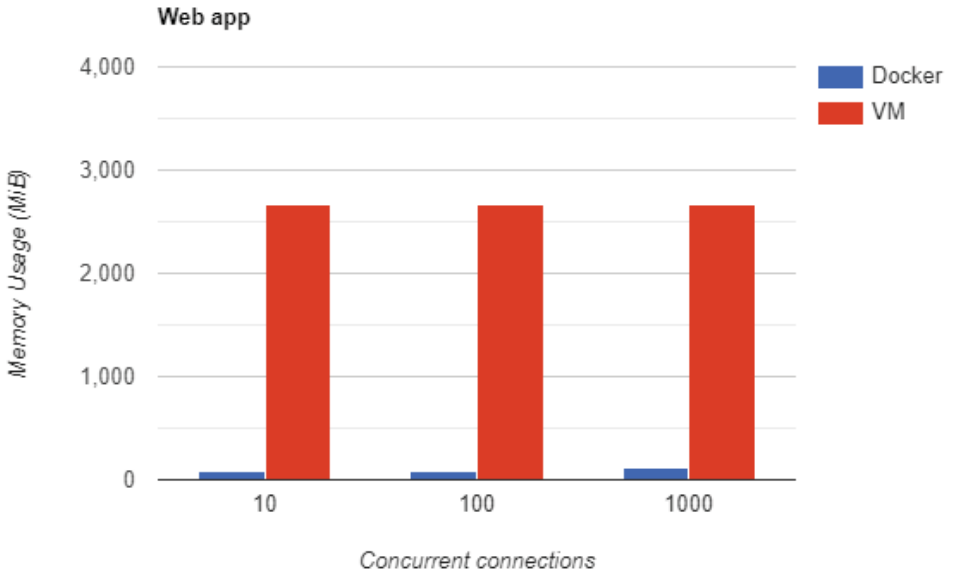

Figure 18 - Memory usage while simulating traffic to the web app

5.3 RQ3: How does the power consumption differ between containers and virtual machines?

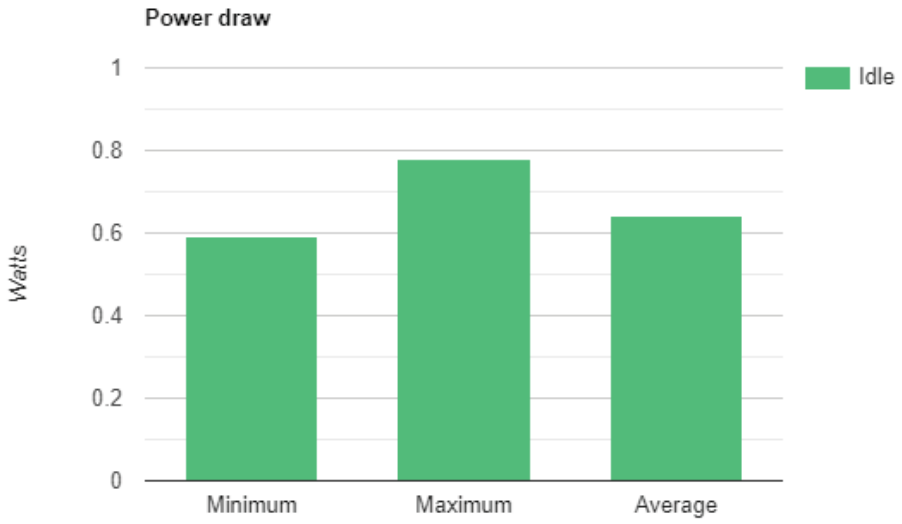

Figure 19 - Power draw of the host machine while idle (no applications running)

Figure 19 shows the power draw of the host machine while no applications are running at all. This was measured to get a base reading of the power draw.

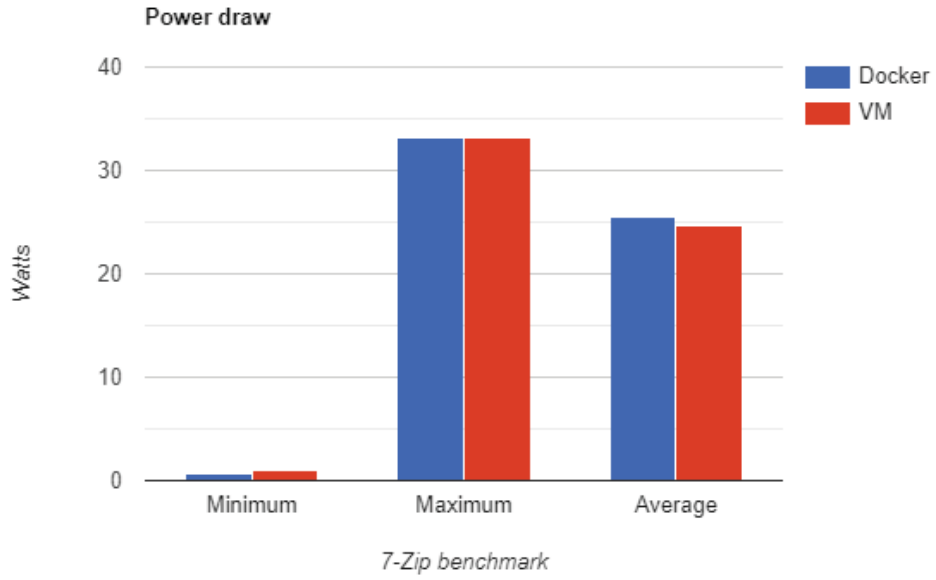

Figure 20 - Power draw of the host machine while running 7-Zip benchmark

Figure 20 presents the power draw results while running the 7-zip benchmark in both a container and virtual machine. This shows that both technologies draw a similar, almost exactly the same, amount of power. The “minimum” shows when the systems are basically idle which was in between test runs, and that’s when the vm was drawing slightly more power.

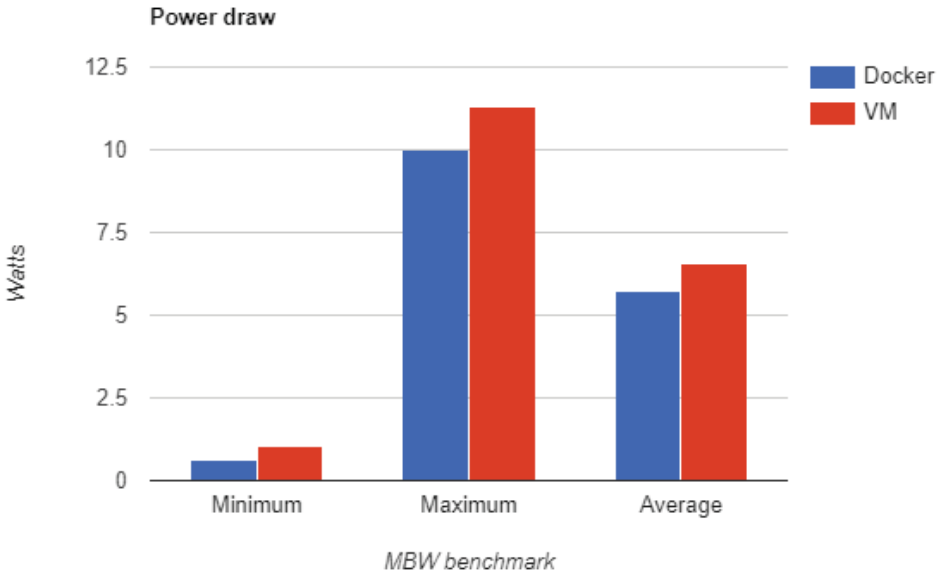

Figure 21 - Power draw of the host machine while running MBW benchmark

As can be seen in figure 21, during the MBW benchmark the vm draws slightly more power than the container, but the difference is only around 1 watt at most.

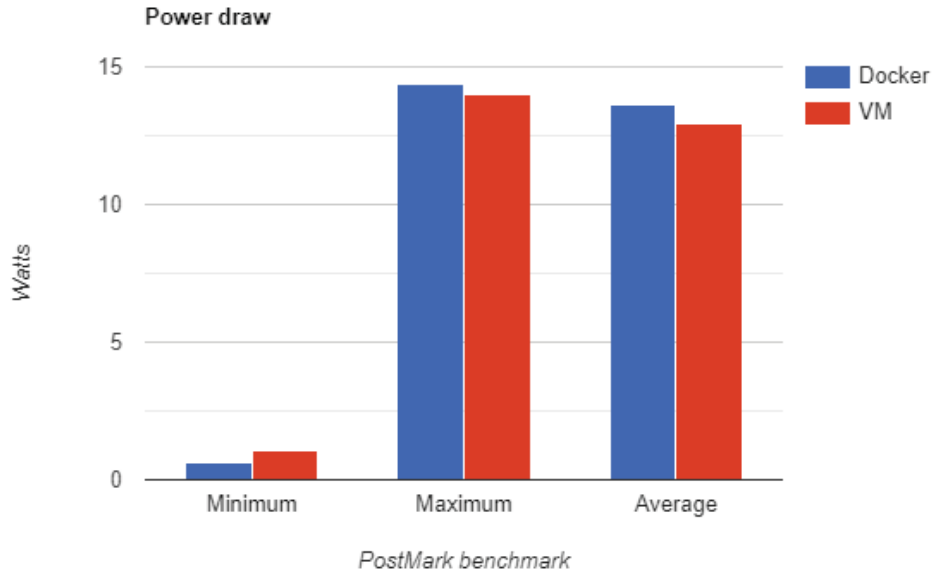

Figure 22 - Power draw of the host machine while running PostMark benchmark

In the PostMark benchmark the container appeared to draw more power than the vm, however this is an extremely small amount, comparable to the amount seen in figure 21.

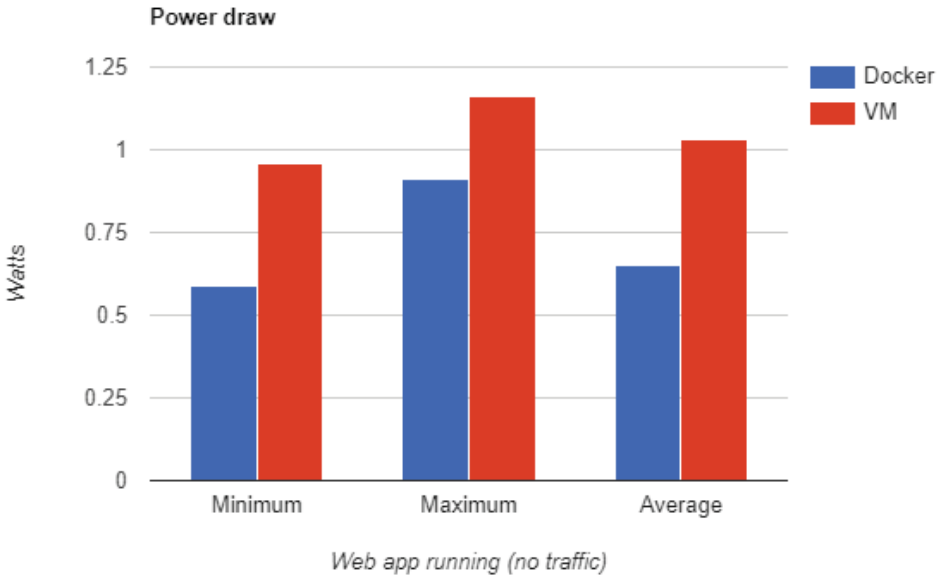

Figure 23 - Power draw of the host machine while running the web application (no traffic)

With just the web application running without traffic, the virtual machine proves to draw more power than the container as seen in figure 23. The difference here, percentage wise, is fairly big but considering that the power difference is not even 1 watt it's not a significant difference at all.

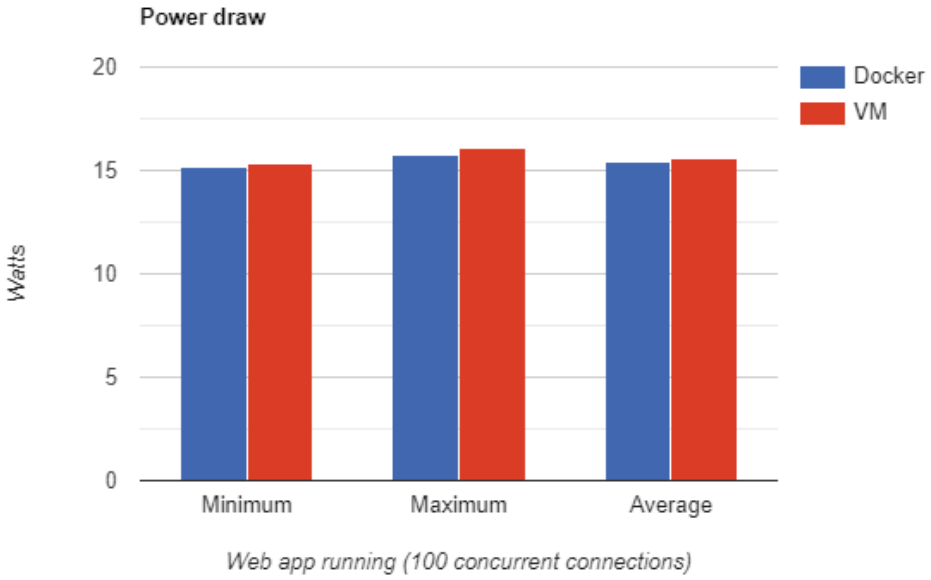

Figure 24 - Power draw of the host machine while running the web application (simulated traffic)

Just like the previous results, figure 24 shows that the power draw is almost exactly the same for the docker container and virtual machine.

Chapter 6

Analysis

RQ1: How does the cpu, memory and disk performance differ between containers and virtual machines?

The results from the literature study showed that every previous study done regarding the performance of containers and virtual machines concluded that containers perform as well or better than virtual machines in all cases for cpu, memory and disk performance. How much faster varied heavily though, as some researchers saw quite a big difference between the two while others didn't really see that big of a difference. The results of A. M. Potdar et al's [5] experiments showed that the docker container was at least two times faster and as much as four to five times faster than the virtual machine in some tests. The interesting part regarding their results are that two of the benchmarks they used were used in our empirical study as well and our results did not match their results at all. For example, one of the benchmarks they used for the cpu was 7-zip compression in which their container performed 8000 MIPS (million instructions per second) whereas the virtual machine only performed 2000 MIPS, making the container four times faster in that benchmark. In our experiments our container performed ~18,000 and the VM ~16,000 MIPS in the compression and both performed equally at ~13,000 MIPS during the decompression, indicating that the container is at most 10% faster than the VM in the same test. The same thing was observed in their memory performance test, in which they used the RAMspeed benchmark test. On average their container achieved 18,000 MB/s while the VM achieved 6000 MB/s marking a 300% difference. Our results of the same test showed that the container achieved ~18,700 MB/s in the add benchmark and ~17,600 MB/s in the copy benchmark and that the VM achieved ~18,800 MB/s in the add benchmark and ~17,500 MB/s in the copy benchmark, making their performance almost identical.

One possible reason for why A.M. Potdar et al's VM performed so much worse than their container could be because they installed their VM on a host machine with windows 10 operating system, which could potentially slow it down significantly if not configured correctly [24]. Although we do not know if this is the case, it could explain why our results differed so much from their results even though we used the same benchmark tests.

W. Felter et al [10] and R. Morabito et al [16] results were more inline with our results, which showed that both virtual machines and containers only introduce negligible overhead when it comes to cpu and memory performance and perform

almost equally. Their results didn't match ours completely though, as in their experiments the container performed as well or better in every test. In our experiments the virtual machine actually beat the container in some tests, albeit by an extremely small margin in most cases. One outlier was the MBW memory copy test where the virtual machine beat the container by almost 30%. We haven't been able to figure out why, as the two other memory benchmarks also performed "copy" operations which resulted in the two technologies performing equally. This has led us to believe that the MBW test results might not be accurate, but we do not really know. Another memory test where the results differed by a big margin was in the tinymembench "memset" benchmark. In this benchmark the container performed about 50% better than the virtual machine. We could not find any information as to what the "memset" benchmark actually does which makes it difficult to analyze and figure out why the difference is so big there.

When it comes to disk performance all the other studies, including ours, agree that containers perform better than virtual machines. W. Felter et al [10] results showed that when it comes to sequential reads and writes, the container and VM perform almost equally but in the case of random reads and writes the VM is almost 50% slower than the container. Oddly enough, R. Morabito et al [16] results show that the VM is roughly 50% slower than the container when it comes to sequential writes and around 75% slower when it comes to sequential reads. We are not sure as to why one study shows sequential reads/writes to be similar between the technologies and one shows a massive difference, however we suspect that this has to do with the fact they used different tests or that one of them possibly wrote "sequential" when they meant "random". Either way the results from both of those studies have the container being faster overall regarding disk performance. In our DBench disk benchmark the container was between 15-50% faster depending on the amount of clients and in the PostMark benchmark the container was around 20% faster than the VM.

RQ2: How does the CPU and memory resource usage differ between containers and virtual machines?

Just like with the performance, the difference in resource overhead introduced by containers and virtual machines seem to vary amongst the studies. Although all of the studies that touched on the topic agree that containers introduce almost no overhead, some studies suggest that virtual machines introduce significant overhead whilst others suggest that they only introduce negligible overhead.

S. Shirinbab et al [9] measured the performance differences along with cpu usage differences while writing to and reading from Cassandra databases in a native non-virtualized environment, Docker containers and virtual machines. Their results showed that virtual machines introduce a significant overhead in terms of resource

usage. One test showed that while performing 40.000 TPS (Transactions Per Second) the native machine utilized ~35% of the CPU while the Docker container utilized ~42% of the CPU and the VM utilized ~60% of the CPU, marking almost a 50% difference in cpu usage between the VM and container. This result does however not match what R. Morabito et al [16] and W. Felter et al [10] found. Although they measured the performance and not the resource usage directly, the performance differences between container and virtual machines being so small indicates that there is no big difference in cpu usage.

We suspect that the reason behind the different results in the studies might be due to using different virtual machines. In our empirical tests we measured both the cpu and memory usage of the host machine while benchmarking either the vm or the container and found that both technologies use an almost identical amount of cpu resources. Both the container and the VM utilized 100% of their available cpu resources during the cpu benchmarks. This equates to 50% of the host machines cpu resources given that both technologies were given 4 cpus out of the hosts 8 available cpus (4 cores with hyperthreading). This along with the fact that they scored similarly in the benchmarks indicates that there is no difference in resource overhead between the two. There was a difference of around 5% at most depending on the test, in some cases the container used more and in some cases it was the VM. As stated in the results, this could also potentially be due to inaccuracies while reading the resource usage as it could fluctuate and be difficult to read during the tests.

While the cpu usage was the same for both technologies the memory usage was not. The virtual machine used significantly more memory than the container. The VM used around 3-4 GiB of memory during the benchmarks while the container used anything between 70 MiB - 2.5 GiB depending on the test. The container only reached ~2.5 GiB of memory usage in two tests, kvazaar (4k) and MBW while the vm used at least 3 GiB during each of them. This is not surprising and is due to the fact that the VM needs an entire guest operating system while the container does not.

RQ3: How does the power consumption differ between containers and virtual machines?

Regarding the power draw differences, all the previous studies resulted in similar conclusions. Congfeng Jian et al [7] compared the power consumption between four different hypervisors, KVM, ESXi, Hyper-V, XenServer along with Docker on different systems and found that none of the technologies consumed more power than any other during the cpu and memory benchmarks. While in some test cases one of the virtual machines drew more power, some other test cases had the containers drawing

more power, though the differences were negligible. R. Morabito et al [17] also compared 4 different virtualization technologies, two hypervisors (KVM and Xen) and two container technologies (Docker and LXC) while running one CPU benchmark, one memory benchmark and one network benchmark. In their approach they used four different test setups, one with only one guest running, one with two guests running, one with four guests running and one with eight guests running simultaneously. Like Congfeng Jian et al, they found that when it comes to cpu and memory load the different technologies draw the same amount of power, however during network load the virtual machines consume more power which was also observed by G. Anusooya & V. Vijayakumar [6].

While S. S. Tadasse et al [18], like the others, found that virtual machines and containers draw the same amount of power during cpu workload and that virtual machines draw more during network heavy loads, they also found that while idle, virtual machines also draw more power than containers.

All of these results seem to correspond to what we found during our experiments, except the fact that virtual machines draw more power during network load since we didn't do any network related tests. Our host machine consumed an average of 0.64W while idle and while running the web application (without traffic) in the docker container the average power draw was only 0.65W. While running the same application under the same circumstances in the virtual machine the power draw averaged 1.03W. Again, this result is similar to S. S. Tadasse et al [18] results regarding idle power consumption. The results from measuring the power draw during the various benchmarks revealed that the two technologies draw a similar amount of power, with the container drawing slightly more in some cases and the VM drawing slightly more in other cases. The difference was miniscule though and at most around 1 watt.

One thing to note about powerstat [20], which is what we used to measure the power consumption, is that it only measures the power consumed by the cpu and not the whole system. Since desktops don't have a battery to measure from, powerstat uses Intel's RAPL interface [25] to do this. This does not seem to compromise the results however, as both R. Morabito et al [17] and S. S. Tadasse et al [18] used external power meters and their results are similar to ours. This seems to indicate that one of the main components in terms of power draw is the cpu.

Chapter 7

Conclusion

In this study we aimed to examine how the cpu, memory and disk performance differs, how the cpu and memory resource usage differs and how the power consumption differs between containers and virtual machines. To answer these questions we conducted a literature study to find out if any previous research had been done, and we conducted a series of experiments of our own. Through our studies and experiments we found the following:

Performance: When it comes to **cpu** performance, we found that containers and virtual machines perform very similarly. There are some small differences in some tests but overall there are no real noteworthy differences. The results of the **memory** performance are a bit more difficult to interpret. Two of our tests had the virtual machine performing better, one test had the container performing better and three had them performing equally. The literature studies showed that containers performed as well or better than virtual machines. Our conclusion here is that they seem to perform similarly for the most part but that depending on the specific memory operations, one might outperform the other. In regards to **disk** performance the container proved to perform better, both in the previous studies and in our own benchmarks. Our experiments had the container performing around 15-50% better depending on the benchmark. These results indicate that virtual machines do not introduce any significant performance overhead when it comes to cpu and memory performance but do so when it comes to disk performance.

Resource Usage: The results of the experiments showed that in terms of **cpu** usage, both the container and the virtual machines used the same amount of resources, with some negligible differences. This result is similar to the other studies except the ones where network load is involved, in which case VMs seem to use more cpu resources. The **memory** usage differs between the two technologies though, as the virtual machine uses significantly more memory. The virtual machine used between 3-4 GiB during all benchmarks while the container used between 70 MiB and 2.5 GiB depending on the benchmark. The reason for the virtual machines heavy memory usage is due to the fact that it needs an entire guest OS.

Power Draw: The results from the studies and our experiments conclude that, while idle, virtual machines draw more power than containers but when put under load they draw the same amount of power. This means that neither technology is a better or worse option than the other when looking at electricity costs or environmental impacts. One exception, according to the other studies, is during network load, where virtual machines consume more power. This is however outside of the scope of our study and not something we tested on our own.

Chapter 8

Validity Threats

In our memory benchmarks we saw some conflicting results regarding the performance differences between the container and the VM. MBW, RAMspeeds “copy” benchmark and tinymembench’s “memcpy” all perform copy operations where data from one memory location is copied to another memory location. While RAMspeed and Tinymembench both had the container and the VM performing similarly, MBW had the VM outperforming the container by roughly 30%. While it seems obvious that there are differences in how the tests work, we do not know what those differences are and why they affect containers and VMs differently. Therefore, this compromises the validity of those results.

The resource utilization measured in our experiments should also be considered a slight validity threat. The reason for this, as stated in the results, is because the readings were done manually. This meant watching the cpu utilization continuously which could fluctuate a bit during the benchmarks. However, we kept track of the highest readings as well as the average usage, so we are confident that the results are accurate with a maximum margin for error of about 5%.

Another aspect that we initially thought could be a potential validity threat was that we used powerstat to measure the power draw of the machine. The reason for this concern was due to the fact that powerstat only measures the power draw of the CPU when doing the measurements on a desktop PC. Because desktops don’t have a battery to measure from, it measures it through an interface available on intel cpu’s. This turned out to most likely not be a concern however, as our results of the power draw differences matched those of the other studies even though they used external power measurement devices.

Another possible validity threat could be that we did not restart the environments after each test. We made sure to give the machines a couple of minutes to “cool down” after each test but we did not restart them. This could potentially cause the resource utilization to be higher during some tests if they were run after other tests.

Chapter 9

Future Work

As can be seen in the results of some of our memory benchmarks, the container and the VM perform equally in two different memory “copy” benchmarks but in another the VM performed roughly 30% better than the container. It would be interesting to investigate what caused this difference to get a more definitive answer to the question regarding memory performance. The same could be investigated for the other categories as well. What is the reason for the container performing 15-50% better in disk benchmarks? These are things that could be examined in future work.

While this study only covered the performance related to cpu, memory and disk of containers and VMs, for future work it would be valuable to also look at the performance differences in regards to the network. Since the majority of VMs and containers most likely handle network requests of some kind, comparing the differences there could be of use.

References

[1] Docker. *What is a container?* https://www.docker.com/resources/what-container/ [Accessed 2022-04-20]

[2] VMware. *What is a virtual machine?* https://www.vmware.com/topics/glossary/content/virtual-machine.html [Accessed 2022-01-24]

[3] VMware. *What is a hypervisor?* https://www.vmware.com/topics/glossary/content/hypervisor.html [Accessed 2022-01-24]

[4] RedHat. *What's a Linux container?* https://www.redhat.com/en/topics/containers/whats-a-linux-container [Accessed 2022-02-25]

[5] A. M. Potdar, Narayan D. G., S. Kengond, M. Moin Mulla, "Performance Evaluation of Docker Container and Virtual Machine", *Procedia Computer Science*, 2020, vol. 171, pp. 1419-1428.

[6] G. Anusooya, V. Vijayakumar, "Reduced carbon emission and optimized power consumption technique using container over virtual machine", *Wireless Network*, 2019, pp. 5533-5551.

[7] Congfeng Jiang *et al*., "Energy Efficiency Comparison Between Hypervisors", *2016 Seventh International Green and Sustainable Computing Conference (IGSC),* 2016, pp. 1-8.

[8] A. Randall, "The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers", *ACM Computing Surveys*, 2021, vol. 53, p. 1-31.

[9] S. Shirinbad, L. Lundberg, E. Casalicchio, "Performance evaluation of container and virtual machines when running Cassandra workload concurrently", *Concurrency Computat Pract Exper*, 2020, vol. 32.

[10] W. Felter, A. Ferreira, R. Rajamony and J. Rubio, "An Updated Performance Comparison of Virtual Machines and Linux Containers", *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2015, pp. 171-172, doi: 10.1109/ISPASS.2015.7095802.

[11] A. M. Joy, "Performance comparison between Linux containers and virtual machines," *2015 International Conference on Advances in Computer Engineering and Applications*, 2015, pp. 342-346, doi: 10.1109/ICACEA.2015.7164727.

[12] S. Singh and N. Singh, "Containers & Docker: Emerging roles & future of Cloud technology," *2016 2nd International Conference on Applied and Theoretical*

Computing and Communication Technology (iCATccT), 2016, pp. 804-807, doi: 10.1109/ICATCCT.2016.7912109.

[13] Preeth E N, F. J. P. Mulerickal, B. Paul and Y. Sastri, "Evaluation of Docker containers based on hardware utilization," *2015 International Conference on Control Communication & Computing India (ICCC)*, 2015, pp. 697-700, doi: 10.1109/ICCC.2015.7432984.

[14] Phoronix. *Phoronix Test Suite* https://www.phoronix-test-suite.com/ [Accessed 2022-02-07]

[15] TechRepublic. *The 10 most popular container tools for businesses* https://www.techrepublic.com/article/the-10-most-popular-container-tools-for-businesses/ [Accessed 2022-02-25]

[16] R. Morabito, J. Kjällman and M. Komu, "Hypervisors vs. Lightweight Virtualization: A Performance Comparison," *2015 IEEE International Conference on Cloud Engineering*, 2015, pp. 386-393, doi: 10.1109/IC2E.2015.74.

[17] R. Morabito, "Power Consumption of Virtualization Technologies: An Empirical Investigation," *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, 2015, pp. 522-527, doi: 10.1109/UCC.2015.93.

[18] S. S. Tadesse, C. F. Chiasserini and F. Malandrino, "*Characterizing the Power Cost of Virtualization Environments*", 2018.

[19] VirtualBox. *Oracle VM VirtualBox* https://www.virtualbox.org/ [Accessed 2022-03-10]

[20] Powerstat https://github.com/ColinIanKing/powerstat [Accessed 2022-03-16]

[21] Apache. Apache Benchmark https://httpd.apache.org/docs/2.4/programs/ab.html [Accessed 2022-03-16]

[22] Wikipedia https://en.wikipedia.org/wiki/Data_compression [Accessed 2022-04-06]

[23] Openbenchmarking https://openbenchmarking.org/suites/pts [Accessed 2022-04-06]

[24] Microsoft https://docs.microsoft.com/en-us/troubleshoot/windows-client/application-management/virtualization-apps-not-work-with-hyper-v [Accessed 2022-04-13]

[25] Intel https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html [Accessed 2022-04-14]

[26] VMware. *What is server virtualization?* https://www.vmware.com/topics/glossary/content/server-virtualization.html [Accessed 2022-04-20]

[27] Zdnet. *What is cloud computing?* https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/ [Accessed 2022-04-21]

[28] Atlassian. *Infrastructure as a service* https://www.atlassian.com/microservices/cloud-computing/infrastructure-as-a-service [Accessed 2022-04-21]

[29] Atlassian. Container vs Virtual Machines https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms [Accessed 2022-04-28]

[30] VMware. *Performance implications of CPU virtualization* https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.resmgmt.doc/GUID-9986F393-4D36-48AB-B839-8FF25ECEEF94.html [Accessed 2022-06-01]

[31] VMware. *Understanding memory overhead* https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.resmgmt.doc/GUID-4954A03F-E1F4-46C7-A3E7-947D30269E34.html [Accessed 2022-06-01]