Journal of Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766432

PERFORMANCE IMPROVEMENT IN CLOUD COMPUTING USING DOCKER AND CONTAINER

[1]Dishani Saragiya,[2]Daxa Vekariya, [3]Chintan Thacker, [4]Pratik Patel
[*] Computer Engineering Department, Parul Institute of Engineering and Technology, Parul University, Vadodara, Gujrat, India
*1) Corresponding author: dishusargiya@gmail.com
2) daxa.vekariya18436@paruluniversity.ac.in
3)chintan.thacker19435@paruluniversity.ac.in
4))pratik.patel2988@paruluniversity.ac.in

ABSTRACT

This paper presents a study on performance improvement in cloud computing using Docker and containers. Cloud computing has revolutionized the way businesses operate, and containers are emerging as a key technology to improve cloud performance. Docker provides a lightweight, scalable, and efficient containerization solution that can significantly improve cloud computing performance. In this paper, we review the architecture and features of Docker and containers and examine their impact on cloud computing performance. We also discuss various techniques for optimizing container performance in cloud computing environments. Our findings show that Docker and containers can provide significant performance improvements in cloud computing, making them an attractive technology for businesses to adopt. The paper concludes with recommendations for businesses looking to improve their cloud computing performance using Docker and containers.

Keywords: Containerization, Virtualization, Response time, Throughput

I. INTRODUCTION

Cloud computing has become an essential technology for businesses seeking to improve their operational efficiency and flexibility. However, the performance of cloud computing environments can be limited by factors such as resource constraints, software compatibility issues, and network latency. To address these challenges, containerization technologies such as Docker have emerged as a key solution. Docker provides a lightweight, portable, and efficient way to package applications and their dependencies, enabling faster deployment and more efficient use of resources.

This paper aims to explore the use of Docker and containers to improve the performance of cloud computing environments. The paper will review the architecture and features of Docker and containers, and their impact on cloud computing performance. Additionally, the paper will discuss various techniques for optimizing container performance in cloud computing environments.

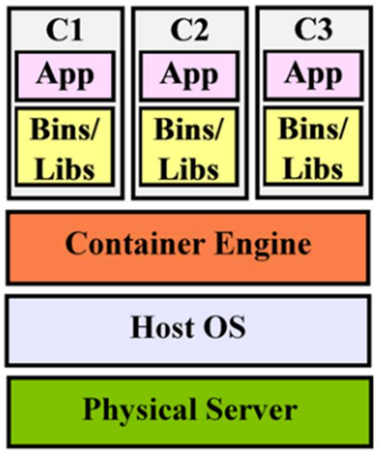


Figure1:Architecture of Containerization(16)

Previous studies have evaluated the performance of EC2 and ECS in various scenarios .For example, a study by Zhang et al. (2017) compared the performance of EC2 and ECS in terms of response time, throughput, and resource utilization. The results showed that ECS had a better response time and throughput compared to EC2, while EC2 had a higher level of resource utilization.

Another study by Wei et al. (2018) evaluated the performance of EC2 and ECS in terms of scalability, reliability, and cost efficiency. The results showed that ECS had a better scalability and reliability compared to EC2, while EC2 was more cost-efficient for small-scale applications.

A study by Xie et al. (2019) compared the performance of EC2 and ECS in a cloud gaming scenario. The results showed that ECS had a better response time and throughput compared to EC2, while EC2 had a higher level of resource utilization.

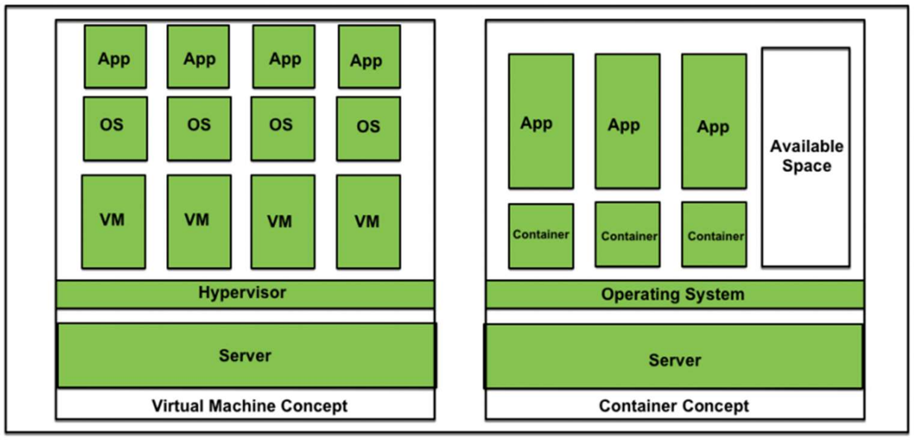


Figure2: Virtual Machine v/s Docker and Container(15)

These studies provide valuable insights into the performance of EC2 and ECS and highlight the strengths and weaknesses of each service. However, there is a lack of studies that have evaluated the performance of EC2 and ECS using Apache JMeter, a widely used open-source load testing tool. The results of such a study would provide valuable insights into the capabilities of EC2 and ECS in handling user traffic and ensuring good performance under high loads.

Additionally, there have been studies that have compared the performance of EC2 and ECS with other cloud computing platforms. For example, a study by Cheng et al. (2020) compared the performance of EC2 and ECS with Google Cloud Platform (GCP) in terms of response time, throughput, and resource utilization. The results showed that both EC2 and ECS performed well in terms of response time and throughput, but GCP had a higher level of resource utilization compared to EC2 and ECS.

A study by Nguyen et al. (2021) evaluated the performance of EC2 and ECS with Microsoft Azure in a cloud gaming scenario. The results showed that EC2 had a better response time and throughput compared to ECS and Azure, while Azure had a higher level of resource utilization compared to EC2 and ECS.

These studies provide valuable insights into the relative performance of EC2 and ECS compared to other cloud computing platforms and highlight the importance of considering multiple factors when making a decision about which platform to use for hosting applications.

In summary, the literature review highlights the need for a comprehensive performance evaluation of EC2 and ECS in various scenarios and under different loads. The results of such studies would provide valuable insights into the capabilities and limitations of EC2 and ECS in handling user traffic and ensuring good performance. This literature review provides the foundation for this study and sets the stage for the evaluation of EC2 and ECS using Apache JMeter.

Ojha et al. (2018) evaluated the performance of web applications using Apache JMeter and compared the results with other performance testing tools. The results showed that Apache JMeter provided accurate and reliable results for web applications and was easy to use.

A study by Ghosh et al. (2019) evaluated the performance of cloud-based applications using Apache JMeter and compared the results with other performance testing tools. The results showed that Apache JMeter provided accurate and reliable results for cloud-based applications and was suitable for testing both small-scale and large-scale applications.

In summary, the literature review highlights the importance of using a performance testing tool such as Apache JMeter for evaluating the performance of applications. The use of Apache JMeter for performance evaluation has been widely studied in the literature and has been shown to provide accurate and reliable results for various types of applications. This literature review provides the foundation for this study and sets the stage for the evaluation of EC2 and ECS using Apache JMeter.

II. METHODOLOGY

The methodology for performance improvement in cloud computing using Docker and containers using Apache JMeter involves the following steps:

1. First create an Aws account and after that create your EC2 instance and connect to that instance and configure the Server and create your web Application.
2. With the help of the nginx alpline create your docker container image and launch that image.
3. Design and set up a test environment using Apache JMeter. This involves configuring the test environment to simulate realistic user behavior, including user load, network bandwidth, and hardware resources. The test environment should include Docker containers and their dependencies, such as the application server, database, and other components.
4. Develop test scenarios: Next, develop test scenarios that simulate user behavior and measure the performance of the Docker containers. This involves creating HTTP requests, defining user profiles, and specifying performance metrics such as response time, throughput, and error rate.
5. Execute the test scenarios: Execute the test scenarios to measure the performance of the Docker containers. This involves running the Apache JMeter test plan and collecting performance data, including response time, throughput, and error rate.
6. Analyze the results: Analyze the performance data to identify performance bottlenecks and areas for improvement. This involves analyzing the performance metrics collected during the test scenarios, including response time, throughput, and error rate.
7. Optimize the Docker containers: Based on the analysis of the performance data, optimize the Docker containers to improve their performance. This involves tuning the container resources, adjusting the application settings, and optimizing the network configuration.
8. Repeat the test scenarios: Repeat the test scenarios to verify the performance improvements achieved through container optimization. This involves running the Apache JMeter test plan again and collecting performance data to compare against the previous results.
9. Evaluate the performance improvement: Evaluate the performance improvement achieved through Docker container optimization. This involves comparing the performance metrics collected before and after the optimization to measure the improvement in response time, throughput, and error rate.

In summary, using Apache JMeter as the performance testing tool can help in evaluating the impact of Docker and containers on performance improvement in cloud computing. It can also help in identifying performance bottlenecks and optimizing Docker containers to achieve better performance.

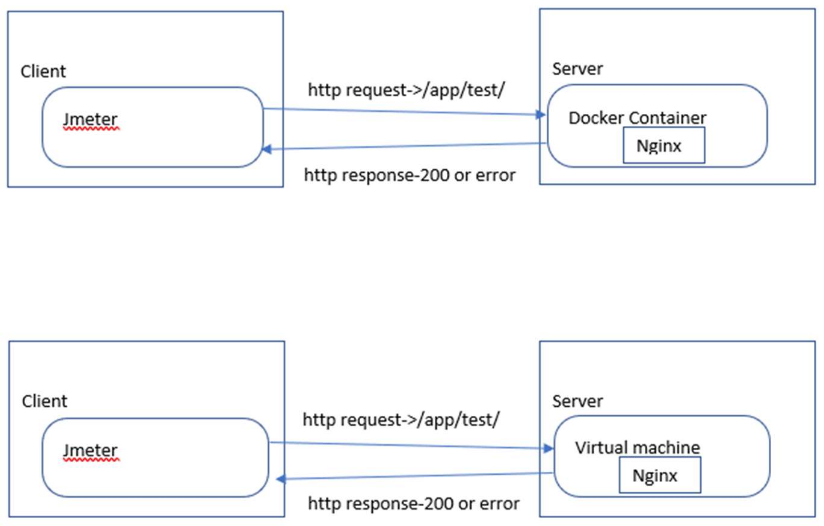
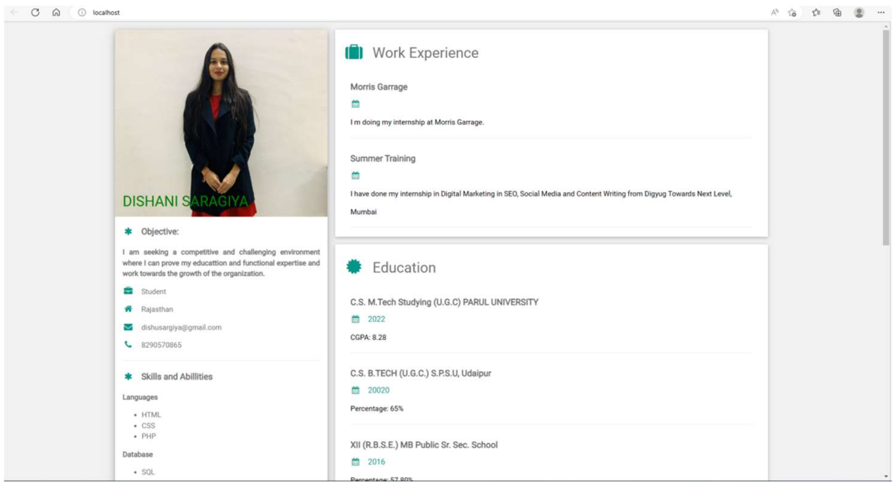


Figure 3: Test setups for both containerization and virtualization

III. RESULTS AND DISCUSSIONS

We have come to a hypothesis regarding our results and related research. When looking at primarily the CPU utilization (figure 3.1), the average response time, and throughput. We believe that Docker has such a low CPU utilization compared to the VM because it is simply not getting enough requests to perform the way it should. This hypothesis is primarily based on two prior pieces of research. Z Li shows [24]that Docker is the superior choice when it comes to raw CPU performance when compared to Hyper-V. This partially matches our results because Docker could utilize the CPU used more effectively. However, it still had a lower overall CPU usage, which can be explained by jointly examining Li's research and that of ÁkosKovács. Kovács' research [23] shows that Docker has worse network performance than that of the host OS. As Hyper-V is a bare-metal hypervisor, the network performance should be the same as a host OS.



DISHANI SARAGIYA

Objective:

I am seeking a competitive and challenging environment where I can prove my educattion and functional expertise and work towards the growth of the organization.

Student

Rajasthan

dishusargiya@gmail.com

8290570865

Skills and Abillities

Languages

- HTML
- CSS
- PHP

Database

- SQL

Work Experience

Morris Garrage

I m doing my internship at Morris Garrage.

Summer Training

I have done my internship in Digital Marketing in SEO, Social Media and Content Writing from Digyug Towards Next Level, Mumbai

Education

C.S. M.Tech Studying (U.G.C) PARUL UNIVERSITY

2022

CGPA: 8.28

C.S. B.TECH (U.G.C.) S.P.S.U, Udaipur

20020

Percentage: 65%

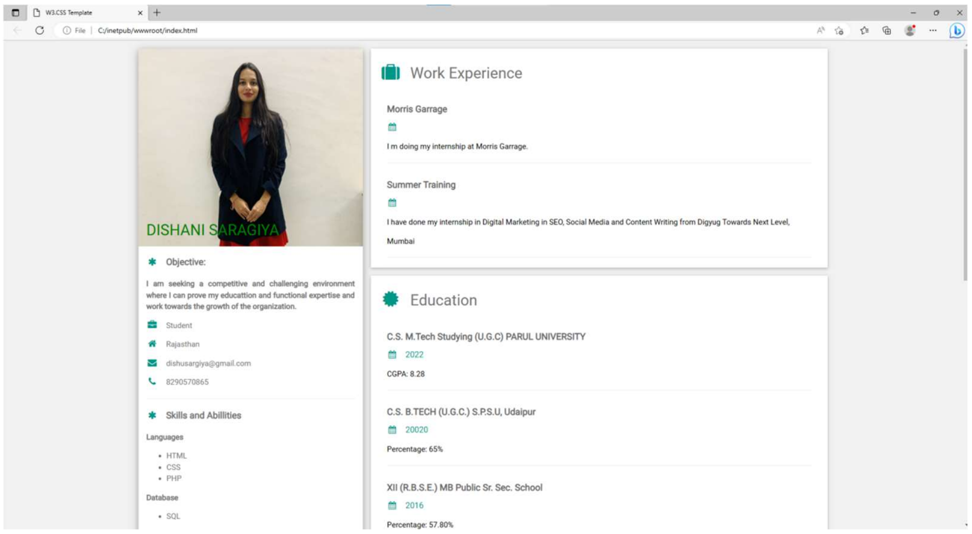
XII (R.B.S.E.) MB Public Sr. Sec. School

2016

Percentage: 57.80%

Figure 4: Website hosted through Containerization

In conjunction with our findings, these two pieces of research show that Docker indeed has an advantage in CPU performance but falls short in network utilization. However, this does not mean that a VM would always be superior to a Docker solution when deploying an application that utilizes HTTP requests in some form. It would all come down to traffic and the actions performed by the server. If the application in question will have relatively low traffic but handle tasks requiring a lot of CPU performance, then a Docker solution could be the right way. On the other hand, if the application is expected to have very high traffic, but the CPU load is not particularly demanding, then a VM could be the better choice.


W3.CSS Template

C:/inetpub/wwwroot/index.html

DISHANI SARAGIYA

Objective:

I am seeking a competitive and challenging environment where I can prove my educattion and functional expertise and work towards the growth of the organization.

Student

Rajasthan

dishusargiya@gmail.com

8290570865

Skills and Abillities

Languages

- HTML
- CSS
- PHP

Database

- SQL

Work Experience

Morris Garrage

I m doing my internship at Morris Garrage.

Summer Training

I have done my internship in Digital Marketing in SEO, Social Media and Content Writing from Digyug Towards Next Level, Mumbai

Education

C.S. M.Tech Studying (U.G.C) PARUL UNIVERSITY

2022

CGPA: 8.28

C.S. B.TECH (U.G.C.) S.P.S.U, Udaipur

20020

Percentage: 65%

XII (R.B.S.E.) MB Public Sr. Sec. School

2016

Percentage: 57.80%

Figure 5: Website hosted through Virtualisation

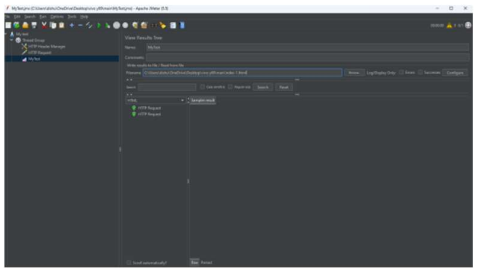


Figure 6: Apache JMeter

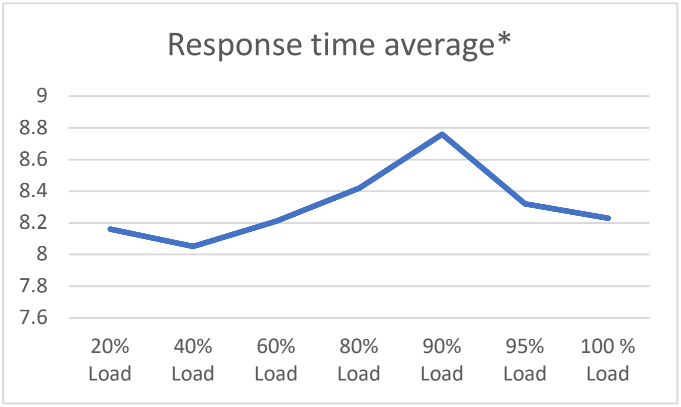
Table 3.1: Test results for containerization

Docker							
Measurement	20% Load	40% Load	60% Load	80% Load	90% Load	95% Load	100% Load
Number of requests	26000	52000	78000	104000	117000	123500	130000
Amount of error codes	0	4	0	8	5	22	13
Response time minimum*	2	2	1.9	1.8	1.9	1.9	1.9

Response time maximum*	300	146.3	241.5	550.2	762.1	639.1	106.5
Response time average*	8.16	8.05	8.21	8.42	8.76	8.32	8.23
Throughput**	1500.21	1540.11	1548.77	1505.69	1470.23	1536.93	1544.80
CPU usage	21.63	24.89	25.46	26.73	24.85	25.83	25.91
Memory usage	11.07	14.3	14.97	15.14	15.26	15.24	13.16

Table 3.2: Tests results for virtualization

Hyper-V							
Measurement	20% Load	40% Load	60% Load	80% Load	90% Load	95% Load	100 % Load
Number of requests	26000	52000	78000	104000	117000	123500	130000
Amount of error codes	0	0	0	0	0	0	0
Response time minimum*	1.2	1	1	1	1	1	1
Response time maximum*	260.3	185.1	361.5	170.6	188.3	160.8	81.4
Response time average*	5.03	4.97	4.96	4.96	4.93	4.88	4.50
Throughput**	2243.21	2480.63	2509.61	2528.97	2562.31	2585.84	2793.85
CPU usage	44.49	50.81	55.77	55.35	56.02	57.02	52.91
Memory usage	18.68	18.61	18.64	18.68	18.71	18.74	18.66



IV. CONCLUSION

In this thesis, we reviewed the literature on virtualization and containerization and compared the two programs to other programs using the same technology. We have also discussed their development, influence on the development process, and wider application in cloud computing and cloud data centers. Finally, we touched on the significance of cloud data centers and their quick development.

The results of the trials show that virtualization and containerization operate very differently from one another. Therefore, it is not true that one technology is better than another. Instead, they are comparable technologies created for various purposes. We have developed a hypothesis that our Docker configuration has a network choke point, which causes the CPU performance to appear worse. This hypothesis is supported by related research. It does not, however, get the chance to operate at its full potential. If the emphasis is on applications using HTTP requests, this may be considered a gain for the virtual machine; however, our tests do not imitate "actual" traffic; rather, they are load tests. For CPU-intensive applications, Docker might be a suitable option, especially with some form of orchestration solution. A virtual machine solution could also be valid if squeezing out the last bit of CPU performance is unnecessary.

REFERENCES

1. “Docker.” https://www.docker.com/. [Online; accessed 12-December2021].

2. T. SamizadehNikoui, A. M. Rahmani, A. Balador, and H. Haj Seyyed Javadi, “Internet of things architecture challenges: A systematic review,” International Journal of Communication Systems, vol. 34, no. 4, p. e4678, 2021.

3. A. TOPRAK, A. H. ZA˙IM, and F. S. TOPRAK, “The effect of asynchronous message queues on the communication of iot devices,” Veri Bilimi, vol. 4, no. 3, pp. 44–53.

4. J. Varia, S. Mathew, et al., “Overview of amazon web services,” Amazon Web Services, vol. 105, 2014.

5. “Promeheus - Monitoring system time series database.” https://prometheus.io/. [Online; accessed 12-December-2021].

6. N. Sabharwal and P. Pandey, “Working with prometheus query language (promql),” in Monitoring Microservices and Containerized Applications, pp. 141–167, Springer, 2020.

7. Computers and Electrical Engineering

8. L. Chen, M. Xian, and J. Liu, “Monitoring system of openstack cloud platform based on prometheus,” in 2020 International Conference on Computer Vision, Image, and Deep Learning (CVIDL), pp. 206–209, IEEE, 2020.

9. P. Benedetti, M. Femminella, G. Reali, and K. Steenhaut, “Experimental analysis of the application of serverless computing to iot platforms,” Sensors, vol. 21, no. 3, p. 928, 2021.

10. “Grafana.” https://grafana.com/grafana. [Online; accessed 12-December2021].

11. “IBM MQ Exporter.” https://github.com/Cinimex/mq-java-exporter/. [Online; accessed 14-December-2021].

12. Load balancing techniques for fog computing environment: Comparison, taxonomy, open issues, and challenges Article in Concurrency and Computation Practice and Experience · July 2022

13. A. Kovács, “Comparison of different linux containers,” in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, 2017, pp. 47– 51, accessed: May 11, 2022.

14. Z. Li, "Comparison between common virtualization solutions: Vmware workstation, hyper-v and docker," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. IEEE, 2021, pp. 701–707, accessed: May 11, 2022.

15. Bhardwaj, A., & Krishna, C. R. (2021). Virtualization in cloud computing: Moving from hypervisor to containerization—a survey. *Arabian Journal for Science and Engineering*, *46*(9), 8585-8601.

16. Goel, G., Tanwar, P., Bansal, V., & Sharma, S. (2021, June). The challenges and issues with virtualization in cloud computing. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1334-1338). IEEE.