THE UNIVERSITY OF WATERLOO
Faculty of Mathematics

The Role of Containers in Modern Application Development

Cannect Home Financing
Toronto, Ontario

Prepared By
Rohit Kaushik
1B Computer Science
ID: 20755173
June 12, 2019

Memorandum

To: Ian Zamojc

From: Rohit Kaushik

Date: June 29, 2019

Re: Work Report: The Role of Containers in Modern Application Development

---

I have prepared the enclosed report on "The Role of Containers in Modern Application Development. This report, the second of four work reports that the Co-operative Education Program requires that I successfully complete as part of my BMath Co-op degree requirements, has not received academic credit yet.

Since the upcoming MVP goal was announced, we have been trying to make Oscar into a cleaner and efficient application. There was talk regarding how tangible it would be to migrate to Kubernetes from Docker, and that got me thinking about how and why containers were created, and what they offer. This report is an in-depth analysis of how containers came into popular usage and what they offer along with what their future looks like.

The Faculty of Mathematics requests that you evaluate this report for command of topic and technical content/analysis. Following your assessment, the report, together with your evaluation, will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers. The combined marks determine whether the report will receive credit and whether it will be considered for an award.

Thank you for your assistance in preparing this report.

Rohit Kaushik

TABLE OF CONTENTS

LIST OF FIGURES

Executive Summary

This report on "The Role of Containers in Modern Application Development" aims to establish how containers rose to importance by examining the history of application/service hosting and comparing them with virtual machines (VMs), an older yet similar construct. It further outlines the future of containers and how enterprises should prepare for this future.

Through careful analysis, the report determines that VMs and containers are closely related and began developing in tandem. While VMs brought about a revolution in hosting services by reducing overhead costs and providing additional security to the enterprise's services, they had drawbacks like increased software maintenance and heavy hardware demands. Thus, their success was limited. Containers, however, provided several advantages like fast development and less maintenance due to their single-OS infrastructure, with support for scaling. The report then suggests some ways in which both individuals and enterprises can prepare for the inevitable shift to a "containerized world" through initiative, logging and testing. Technology moves forward, and the report considers what comes next for container technology – including a decentralized orchestration system.

Finally, the report concludes that while containers will dominate the hosting space for some time, it faces vital issues with security for multitenant infrastructures. Consequently, researchers and scientists will have to look back on the lineage of containers to assess trade-offs made along the way and figure out ways to take the technology forward.

1.0 Introduction

Back when the internet was created, the first websites and apps were hosted by enterprises on local servers (Randall, 2019). When they had to pick the architecture of the server, naturally, they chose the best option – big and fast. Therefore, if a company had many services, or had to scale their existing service(s), they had to utilize multiple of these 'big' servers, since each server could only host one service and manage only a certain amount of traffic load. They were and often still are placed in sites known as "Data Centres." However, despite offering a means to host apps and websites, this system had inherent disadvantages. Multiple servers meant more storage space, more electricity, more maintenance costs, more licencing costs and risks of shutting down in case of power outages or physical damage due to weather and other similar factors (Randall, 2019).

Soon, researchers found a way to mitigate the costs incurred due to having multiple servers in data centres – Virtual Machines (VMs). Now, the server's resources like memory and RAM was divided up into certain number of parts, and each part was equipped with an Operating System, a VM and the app/service itself (Noble et al., 2006). In this manner, multiple apps/services could be hosted on one single server. According to Noble et al. (2006), this separation provided security to the system as services only had to work with a small VM environment. Despite the attractiveness of this invention and the incredible market disruption it created, VMs had their own downside. To support the multiple Operating Systems on each server, more licences and anti-virus software had to be purchased, installed and maintained. Furthermore, there had to be an administrator for each Operating System to ensure smooth running and to oversee updates.

Thus, VMs almost eliminated data centres and reduced the world's carbon footprint dramatically, but enterprises were still directing a lot of money towards hosting their services (Codd, 1962).

Finally, in 2008, Linux came out with their Containers (LXC). Containers are essentially an improvement on VMs, wherein only one Operating System is installed on the server (Wright et al, 2002). The Operating System itself is divided into parts and each part can run an application. This reduced additional Operating System related costs and is currently an increasingly popular tool to host services, which an increasing number of enterprises are adopting each day. This report further goes into detailing the comparison between VMs and Containers, explaining how containers have brought about a massive change in the world of technology. It sheds light on the interplay between cloud computing and application hosting. Furthermore, it explores the future of Containers, what that means for technology and enterprises, and how they can prepare for the upcoming change.

2.0 Analysis

In order to truly understand the necessity of Container technology in the current landscape, their very successful predecessors, Virtual Machines, need to be thoroughly analysed. After understanding their working and shortcomings, we can begin to truly appreciate the need for Containers. Since this technology has just kicked off, it will be worthwhile to explore how enterprises can prepare to harness all the benefits of Container technology. However, in technology, change is the only constant. Thus, analysis of the shortcomings of Containers will open avenues for future development.

2.1 The Virtual Machine Revolution

Both VMs and containers originated from a "shift in hardware [and] software architectures in the 1950s" (Codd, 1962). VMs were deployed first, after newer hardware introduced 'multiprogramming', which included multitasking in the form of simple context-switching and multiprocessing in the form of related I/O processors and multiple CPUs. Here, the physical server has VMs, each with an OS. By running several of these, the server's resources get consumed fast, and only a few VMs can be run (DeMuro, 2018; Fig. 1).

According to Randall (2019), multiprogramming increased software complexity and made misbehaviour more likely, since processes were now isolated from one another and ran on separate VMs in the server. However, it was found that this software isolation was in fact a requirement to prevent one process from making changes to another. The solution to this isolation was giving the kernel access to the underlying hardware to manage the parallel

processes. Opler and Baird (1959) argued that this approach has the potential to improve and provide portability for programs not written for parallel processing. Madnick et al. (1973) saw that the portability of virtual machines was an advantage for developers, allowing testing of many versions of the operating systems in different artificial environments, but on the same computer. Along with this, enterprises did not need a multitude of gigantic servers anymore, reducing both their cost and the global carbon footprint.

However, as time and technology moved on, companies like HP and Intel brought smaller processors to the market (1970s) which could not support virtual memory, and thus underwent the decline of VMs (Creasy, 1981).

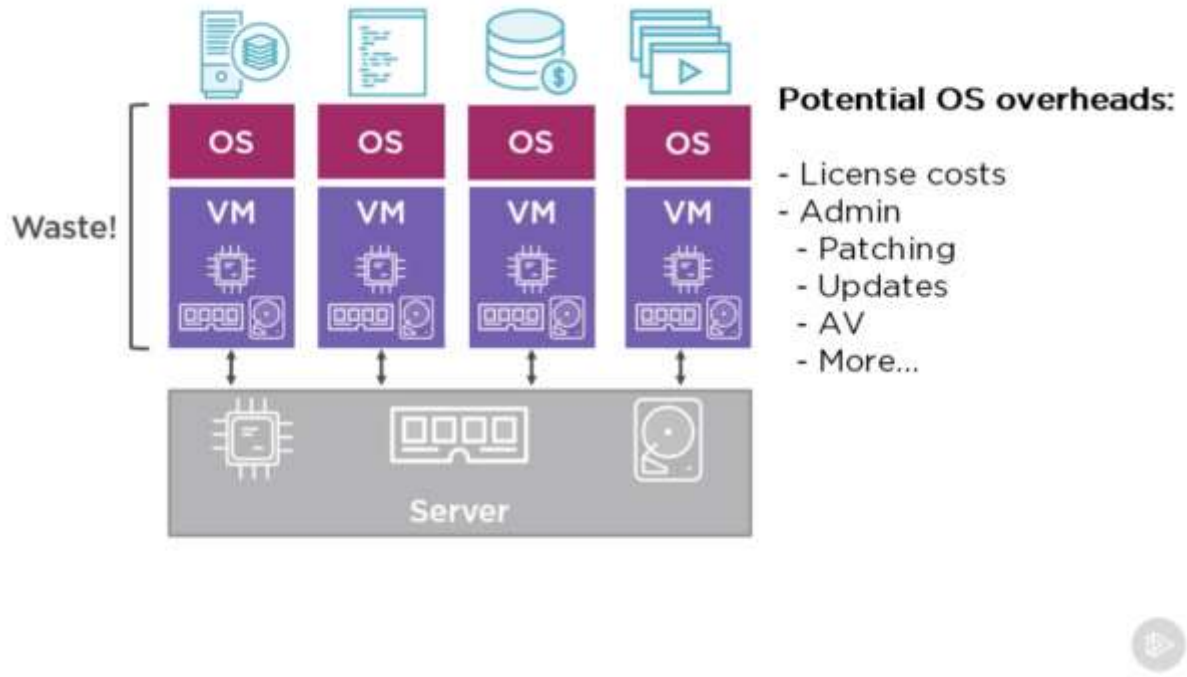

Fig. 1: VM infrastructure and its overheads (Poulton, 2019).

2.2 Containers in today's Tech Landscape

In containers, the server has one OS; containers are built onto this singular OS and they "share its resources" (Wright et. al, 2002). For security, the shared parts cannot be accessed by the process itself. Thus, containers use more of the OS' resources than the server's, so more containers can be packed onto a single server than VMs (DeMuro, 2018; Fig. 2). Each package is called a microservice. This is advantageous as teams can work on each of the containers separately while maintaining the relationship between the components, which accelerates software development (DeMuro, 2018). For example, in a survey conducted by Bain & Co. (2017) of 449 U.S. executives and leading enterprises, adopters reported 15% to 30% reductions in development time. Adopters also reported initial cost savings of 5% to 15% due to lesser server and hardware costs (Taylor et al., 2017).

Containers are paving the way for digital transformation. Container technology is easily portable, which improves the flexibility and scalability of IT architectures. To encourage this portability, enterprises will be encouraged to move their services on the cloud, thus promoting cloud-computing in general (DeMuro, 2018). VMs need several minutes to start since it is basically a miniature computer, just like the PC on your desk. However, containers can be started in a few seconds by simply running a command. In this way, containers can be scaled during high traffic and downscaled during quieter times. If they crash, they can be restarted right away. Today, there are several tools like Docker Swarm and Kubernetes which are container orchestration tools. Essentially, they manage container clusters and decide when to upscale, downscale, spit up more containers, shut them down or restart them depending on health and network traffic (Poulton, 2019).

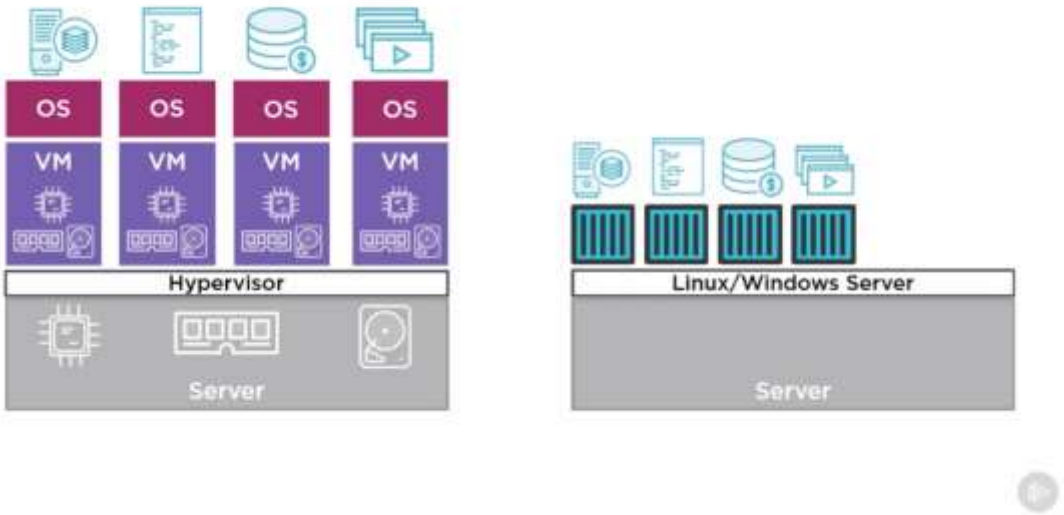

Fig. 2: Comparison of VM and Container Infrastructure (Poulton, 2019).

2.3 Preparing for a Containerized World

Based on performance evaluations of container technology by Xavier et al. (2013, 2014), we can extrapolate the following conditions for which results will likely be affected by the implementation of the underlying OS-level virtualization. Enterprises looking to move to a containerized model for their services should be mindful of the following:

i. Memory bandwidth is of significant importance (i.e. if 5% of performance will affect results).
ii. Network address translation (NAT) is required.
iii. Distinct projects consolidated on the same host.
iv. For a project, containers with conflicting roles need to be co-located in the same physical host (e.g. two database instances on the same host).
v. Kernel version can't be frozen. Any project for which any of the above applies should be carefully examined since the effects of containerization can affect the results.

Poulton (2018) believes that to successfully thrive in a “containerized world”, it is important for both ***individuals*** and ***enterprises*** to be well prepared:

i. ***Individuals*** must combine both knowledge and experience to grow. He recommends playing around with technologies like Kubernetes and Docker thoroughly before contributing professionally. This contributes to experience. Watching videos, doing courses and reading research papers are the best sources to gain more knowledge on containers.

ii. ***Enterprises*** should consider the reality where they run their services on containers. It is important to do so because containers are soon going to be the go-to technology for hosting, and enterprises should, in the early stages, consider how containers might boost their business. If they find this useful, management should take initiative and push developers to migrate to containers, while ensuring that they keep tabs on the continuous integration and continuous delivery process. They should begin by hosting parts of their application as containers, and only once they verify that it works, should they migrate their whole application. They should also invest in reliable monitoring, logging and orchestration tools to keep tabs on the functioning of the containers and the whole application.

2.4 Looking Ahead

Since multiple containers share the same OS, DeMuro (2018) believes that there are concerns regarding the security of the whole system. If the OS is compromised, all the containers will be too. As mentioned above, containers can rapidly scale containers. This can

create troubleshooting problems in versatile environments. If there is a problem or a bug hidden, it will be scaled and spin out of control. Around 1/4th of companies run 10 or more containers simultaneously on a single system. In a container environment, updates and repairs must be multiplied by the number of containers—and the sheer volume of units may require a heightened level of management capabilities and additional resources. If an enterprise's asset management system cannot cope with the monitoring and diagnosis of machines, containers will spin them out of control.

This is something scientists and developers will have to work on to truly drive successful container technology. Since VMs are much more secure, they are quite often used in tandem with containers today; this will continue until containers become as secure as their counterpart VMs (Randall, 2019). Research is also being done to decentralise the container manager server to improve security (Poulton, 2019).

3.0 Conclusion

The report finds that due to the development of hardware, multiprogramming was introduced where a server's resources can be divided which led to the creation of VMs. They provided a cheap and efficient way to host services, but still led to high maintenance and overhead costs. Eventually, smaller processors meant VMs could no longer be fully supported. Soon, containers were developed which significantly reduced the overheads posed by VMs and worked on modern hardware.

By containerising applications, teams could individually work on and scale parts of the

application while maintaining the overall structure. This led to faster development time. With containers, the option of quickly spitting up instances was available as compared to slow VM booting time. This technology also encouraged cloud computing. Since containers are going to be a prevalent means to host applications, enterprises should consider using them to boost their business; they should keep in mind precautions like memory bandwidth, NAT testing, logging, testing and maintenance while performing migrations. Individuals should also be properly equipped with knowledge and experience with container technology before contributing professionally.

While it offers several advantages, container technology also faces security issues as containers share an underlying OS. Research is being done to build a decentralised server system that hosts containers to alleviate this problem.

Acknowledgements

This report was inspired by conversations with colleagues regarding hosting apps on Containers (specifically, Kubernetes) and a challenge given by them to deploy their application on Kubernetes containers. It was created through thorough research using recent and slightly older scientific papers authored by professors and scientists involved in the innovation of Containers and VMs.

References

1. *A. Opler and N. Baird (1959). Multiprogramming: The Programmer's View. In Preprints of Papers Presented at the 14th National Meeting of the Association for Computing Machinery, ACM '59, pages 1–4, New York, NY, USA.*
   From - doi:10.1145/612201.612215

2. *A. Randall (2019). The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers. University of Cambridge.*
   From - arXiv:1904.12226v1

3. *C. Wright, C. Cowan, S. Smalley, J. Morris, and G. Kroah-Hartman (2002). Linux Security Module Framework. In Proceedings of the Ottawa Linux Symposium, pages 604–617, Ottowa, Canada.*
   From - doi:10.1109/fits.2003.1264934

4. *DeMuro, Jonas (2018). What is Container Technology?*
   From: https://www.techradar.com/news/what-is-container-technology

5. *E. F. Codd (1962). Multiprogramming. In F. L. Alt and M. Rubinoff, Advances in Computers, volume 3, pages 77–153.*
   From: doi:10.1016/s0065-2458(08)60618-x

6. *J. Taylor, P. Renno (2017). What Role Will Container Technology Play In The Digital Journey? para 7, Bain & Company.*
From: https://www.forbes.com/sites/baininsights/2017/06/19/what-role-will-container-technology-play-in-the-digital-journey/#7c2b37a73c07

7. *M. Xavier, M. Neves, F. Rossi, T. Ferreto, T. Lange, and C. De Rose (2013). "Performance evaluation of containerbased virtualization for high performance computing environments." 21st euromicro international conference on parallel, distributed and network-based processing (PDP), pp. 233–240.*
From – doi:10.1109/ic2e.2015.75

8. *M. Xavier, M. Veiga Neves, and C. Fonticielha de Rose (2014). "A performance comparison of container-based virtualization systems for MapReduce clusters." 22nd euromicro international conference on parallel, distributed and network-based processing (PDP), pp. 299–306.*
From – doi:10.1109/ic2e.2015.75

9. *N. Poulton (2019). Docker and Kubernetes: The Big Picture.*
From - https://app.pluralsight.com/library/courses/docker-kubernetes-big-picture

10. *Noble, Brian and Chen, Peter (2006). When Virtual Is Better Than Real. Department. of Electrical Engineering and Computer Science, University of Michigan.*
From – doi:10.1109/hotos.2001.990073

11. *R. J. Creasy (1981). The Origin of the VM/370 TimeSharing System. IBM Journal of Research and Development, 25(5):483–490.*
From - arXiv:1904.12226v1

12. *S. E. Madnick and J. J. Donovan (1973). Application and Analysis of the Virtual Machine Approach to Information System Security and Isolation. In Proceedings of the Workshop on Virtual Computer Systems, pages 210–224, New York, NY, USA.*
From - arXiv:1904.12226v1

13. *W. Felter, A. Ferreira, R. Rajamony, and J. Rubio (2014). "An updated performance comparison of virtual machines and linux containers," technology, vol. 28, p. 32.*
From – doi:10.1109/ic2e.2015.75

14. *X. Tang, Z. Zhang, M. Wang, Y. Wang, Q. Feng, and J. Han (2014). "Performance evaluation of light-weighted virtualization for PaaS in clouds." Algorithms and architectures for parallel processing, X.-h. Sun, W. Qu, I. Stojmenovic, W. Zhou, Z. Li, H. Guo, G. Min, T. Yang, Y. Wu, and L. Liu, eds., Springer International Publishing, pp. 415–428.*
From – doi:10.1109/ic2e.2015.75

Image Sources *

1. *Fig. 1: VM infrastructure and its overheads.*
From - https://app.pluralsight.com/library/courses/docker-kubernetes-big-picture

2. *Fig. 2: Comparison of VM and Container Infrastructure.*

   From - https://app.pluralsight.com/library/courses/docker-kubernetes-big-picture

* The Images are screenshots from videos on Pluralsight.