Advance Container Consolidation In Cloud Using Time Selection Algorithm

MSc Research Project
Cloud Computing

Nitin Fernandez
Student ID: X18200346

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nitin Fernandez
Student ID:	X18200346
Programme:	Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	Advance Container Consolidation In Cloud Using Time Selection Algorithm
Word Count:	9841
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Advance Container Consolidation In Cloud Using Time Selection Algorithm

Nitin Fernandez
X18200346

Abstract

Cloud Computing has gained wide spread attention in the IT world as it has changed the way we operate on computing and provided us with solution for automatically managing resources and also provided users with modern service delivery models. Container based clouds have been a new model launched by various cloud providers that enable applications that are deployed on them to run in an isolated run-time environment where the the OS resources are shared between them. Also containerized application gain certain advantages from running in containers which are scalability, isolation, and are highly portable. The increase demand in cloud computing resources and management of cloud resources has lead to adoption of a new paradigm of computing which is sustainable cloud computing where resource management is a key factor as such we notice a wide adoption of container based cloud as they experience none of the performance overheads experienced by traditional hypervisor based virtualization the study compares the performance between virtual machines and container. While also looking into more resource aware approaches in cloud which is consolidation for more efficient management of resources. This study simulates a novel consolidation based algorithm using CloudSim where it is compared to the current state of the art approach keeping parameters such as number of migrations and resource consumption.

1 Introduction

Cloud computing is a technology that has changed the entire paradigm of the computing environment it is a utility oriented delivery of computing resources to any user located anywhere in the world using a pay as you go model. The main stream of cloud service are as SaaS- software as a service, Paas- Platform as a service, and IaaS- Infrastructure as a service. Cloud computing has enabled many companies to discard their on-premise infrastructure and opt for virtual resources instead of physical resources as there is no shipping and time constraints on these resources and virtual resources can be deployed as soon as they are purchased from the cloud vendor. Most cloud environments have been build on the foundation of hypervisor based virutalization this techniques has been efficiently able to consolidate cpu from one machine to another virtual machine but it cannot consolidate memory as efficiently from machine to machine. But in-respect to this a new solutions called Container based virtualization has come which is a more lightweight approach than traditional hypervisor based approach. Gill and Buyya (2018) Container don't experience the same performance overheads as that of hypervisor based virutalization they exhibit performance to that of near native which is what most futuristic applications like HPC, IoT, and Embedded applications require. Any application that can be hosted on a common OS can be deployed in containers.The main advantage that container provide is its performance which is almost similar to that of bare-metal some of the other properties container exhibit are resource sharing, isolation, and dependibility. Container share resources from the host operating system also their hardware resources with the guests deployed on them. The resource utilization depends on the application running in the container. There is greater isolation in container than in VMs because of the shared OS with the host. Often large distributed systems are shared in container which has developed a trust model between applications running in the same container so that there myabe no conflict or resource contention between the applications this model ensures dependability. Many researchers have suggested the container based approach could replace hypervisor based approach as the base platform for building the new cloud environment. Felter et al. (2015).

The increase in demand of cloud resources has seen cloud providers adopt new resource management solutions and also a sustainable cloud computing approach to resource management this new paradigm makes it important for cloud providers and cloud user to be able to quantify the performance. The rise of this new paradigm of computing has given rise to adopting more lightweight approach's of virtualization. This saw too the containerization of applications, microservices, serverless computing, and SOA which are the latest in the domain. Containers have become the latest trend in comparison to traditional virtualization technologies containers run as a process that enables the virtualization environment to reduce overall performance overheads. Gill and Buyya (2018)

1.1 Container As A Service

Containerization of applications have gained a wide spread attention in the industry as the service provides the application running in the container with many additional benefits like scalability, portability, flexibility, agility and an isolated run time environment. PaaS provides developers with a platform to deploy their code without worrying about the underlying infrastructure and hardware despite this advantage there is still an issue of the applications deployed on Paas being platform specific this is where CaaS comes into play where they provide the application with an isolated runtime environment where they application runs as a process. Container increase the performance and efficiency of cloud resource utilization. The major players that have adopted this approach of CaaS are Amazon with their service of ECS- Elastic Container Service and Google Container Service. Most CaaS service are deployed on top of existing IaaS or on Virtual Machines as the major players argue that deploying containers on VMs add an additional layer of isolation and security but some researchers have also question this approach of deploying container on VMs Felter et al. (2015). But by running containers on bare-metal the users cannot run mixed workloads on containers but that is not the case for virtual machines. CaaS provides portability of applications which overcome the platform dependencies of applications. The increase demand of cloud services has led most cloud providers to look into resource management principles to obtain more sustainable computing power. VM consolidation has been a consolidation technique of ideal virtual machines which has been quite successful although there is huge performance degradation in terms of startup and shutdown time of Virtual machines with the rise of CaaS most providers are looking at better consolidation strategies for consolidation in container for better resource management although container do not experience the same performance degradation as virtual machines.

1.2 Research Problems and Objectives

This thesis looks at the resources management problem in container based clouds. In summary the following research problems are explored.

1.2.1 What is the research question?

What methods can be considered to improve workload/resource efficiency during the container consolidation process?

1.2.2 What is the current issue?

The major issue with the current system is that Hypervisor based virtualization has huge performance overheads when it comes to dealing we with HPC, Hadoop, and IoT workloads hence we look to a suitable alternative to hypervisor-based virtualization which is container based virtualization. Although this technology is at its infancy it has great potential to for the new standard to build the next paradigm of the cloud environment. Most futuristic technologies like IoT, Hadoop, and HPC require near native performance which can be achieved by container based virtualization as it provides these new technologies with an isolated run time environment, portability, and almost no performance overheads. Most applications deployed on PaaS service are platform dependent but this problem is solved with containerization where CaaS or container based clouds come into focus but with the huge demand of computing resources there has been a change in paradigm to sustainable cloud computing where it is important for cloud providers and users to understand the difference in performance implications.

1.2.3 Objectives

- **LO1** We look to benchmark the performance of traditional hypervisor based virtualization to that of more lightweight OS-level virtualization. We choose KVM as they hypervisor and Docker containers as the OS-level virtualization of choice.

- **LO2** We look at metrics and tools to provide a fair analysis of our benchmarking process of both environments
- **LO3** We implement a novel consolidation time algorithm to improve the consolidation process.
- **LO4** We simulate the algorithm in container based clouds for efficient container consolidation and compare its performance and resource management policy to that of traditional VM consolidation under the same policy.

1.3 Organization of Research

- **Section2-** This section provides us with an overview of related works as well as the literature review of substantial research material by fellow researchers in the domain.
- **Section3-** This section describes the adopted methodology used to conduct the research for the proposed work.
- **Section4-** This section gives us a clear idea and picture of the design specifications on the proposed work to give us a clear road map for the implementation process.
- **Section5-** This section gives us the detailed workout of our implementation of the proposed work along with all tools and variables used for our work.
- **Section6** This sections provides us with the evaluation and explanation of results obtained from the conducted experiment.
- **Section7-** This section provides us with the final conclusions to this thesis and also entails a possible future road map to the research that can be conducted in the domain.

2 Related Work

As part of the research project we give an brief overview of our project proposal and its main goals to be researched as part of this project. Cloud computing has been the the latest technology to grow at a rapid pace in the current paradigm of technology although there is wide scope of adoption of this new technology in the form of Network ,Storage, and Compute as the main form of services. This wide scope of adoption of cloud technologies has given rise to a new taxonomy of computing which is sustainable cloud computing. This enables the optimum use of cloud resources to ensure reduction in resource consumption for better efficiency. For over a decade now Hypervisor-based virtualization has been the defacto standard for building any cloud environment. Although this seems like a very successful option in running most applications, It fails when it comes to futuristic technologies such as Hadoop , IoT, and HPC workloads where it produces performance overheads as such the performance of these futuristic technologies are affected on the whole hence we require a viable alternative for these technologies as found from various research paper we find that container are a clear alternative to hypervisor based virtualization as they produce almost no performance overheads and performance of bare-metal. They also provide these technologies with additional features like an isolated run-time environment and enable application portability.This has given rise to a new paradigm of computing of containerization of applications and microservices. Although this technology is still in its infancy it has great scope to take over as the next basis for building the new cloud environment as new futuristic technologies require near native type of performance.

2.1 Literature Review

2.1.1 Container-based Virtualization Vs Hypervisor-based Virtualization

As part of the research proposal vast research was done in to provide sufficient evidence to state that Container based virtualization can run futuristic applications without the same performance overheads observed in Hypervisor based virtualization. As such we first look at the taxonomy of sustainable cloud computing which gives us a brief idea of the concept of optimization in resource utilization can greatly enhance the productivity of the current system by adopting more lightweight approach of virtualization with less performance overheads. Some of the main questions that the taxonomy of sustainable computing poses are allocating of resources in a efficient manner, which node can be turned off for improved energy consumption, when is the right time to migrate workloads from one machine to another to improve efficiency. We then compare the architecture of docker container to that of VM's and show that with docker the applications can achieve an isolated run-time environment and also makes it portable . Also we notice that server based virtualization requires an OS to be loaded onto each VM which after sometime becomes a tedious task which was one of the main reasons for the development of containerization.

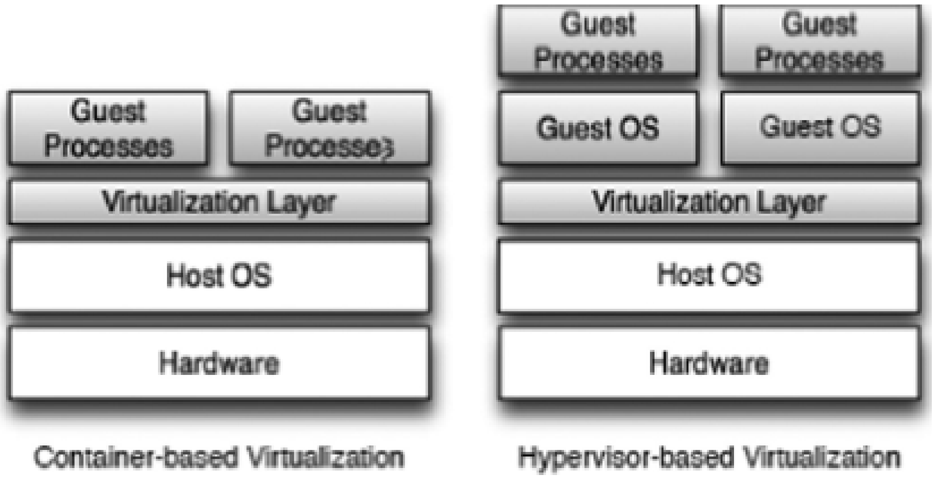


Figure 1: Native vs Virtualized
Li et al. (2017)

2.1.2 Virtual Machines Vs Containers Performance Analysis

Wes et al conducted a study to show the performance of VMs to that of linux containers. The study chooses KVM as the Hypervisor and Docker container as the container for the experiment. The study also chooses MYSQL at the database to run workloads as it has been shown to stress memory, network, file system. The study employs the use of sysbench as the becnhmarking tool to evaluate the memory and CPU overheads. The study concludes by showing that container experience almost negligible overheads for memory and cpu for I/O intensive workloads. The study also questions the practice of deploying containers inside of VMs as they cause performance overheads as such they suggest deploying directly on virtualized environment.Felter et al. (2015) Xiao et al in his study states that traditional virtualization like KVM, Xen are typically via the host OS to build the guest OS. The guest OS requires a lot of resources which is a waste of available resources where as containerized applications share available resources. The experiments aim to benchmark the cpu utilization and disk i/o performance using a benchmarking tool Bonnie++. The study concludes that containers experience efficient resource consumption in comparison to VMs.Xie et al. (2018) Zanibhek et al provides us with a study that benchamrks the performance comparison of various container-based technologies. The study states that container-based technology is over-taking hypervisor based virtualization. The study conducts experiments using various benchmarking tools to evaluate the performance of CPU utilization, disk I/O. The CPU performance is benchmarked using Y-crusher and linpack as benchmarking tools while disk i/o performance is evaluated using Bonnie++ benchmarking tool. The study depolys the containers on NecTAR an research based cloud environment using ubuntu 16.04 image for the experiment although this was done to rule out any discrepancies in results between a standard setup and that of cloud environment which shows the same results on either environments. The study concludes showing that container based virtualization has no overheads in terms of CPU performance and Disk I/OKozhirbayev and Sinnott (2017). Zheng et al also provides a study for the performance comparison between Hypervisors and containers although there have been many studies containers did not get popular until the launch of docker. The study also proves IBMs study Felter et al. (2015) to be in-accurate as it denies IBM's report of containers and VMs experiencing no overheads on CPU and memory usage. The study also shows the containers do experience overheads but not to the extend of that of VMs and hence in-comparison container-based approach is a more light weight solution.Li et al. (2017) Walter et al provides us with a study to compare the virtualization technologies for HPC applications and workloads we mainly focus on OPENVZ and XEN as the two main areas of comparison for this study. OpenVZ being a linux based container while XEN is a famous hypervisor of choice for most studies. Virtualization is a common technology for improving utilization of resources but in terms of HPC applications and workloads. The following study employs NPB MPI benchmarking tool to evaluate the performance overhead of OpenVZ and Xen for a MPI workloads but the following study only checks for CPU utilization and performance for MPI workloads which is just a single aspect of this test. The study determines although OpenVZ did not perform to that as bare-metal performance it still performance close to that of bare-metal and displayed significantly lower overheads that XEN hence we can see that the study also shows container-based virtualization to show better performance in terms of workloads.[6] Nathan et al provides us a study with where he states that the benefits of virtualization are server consolidation,isolation and availability but with the HPC workloads and futuristic technologies there is huge need to maximize throughput without loss of cpu utilization and I/O efficiency. The study compares KVM and OPENVZ for HPC environments although OPENVZ does not perform to that of bare-metal it still performance as close to near native performance and much better when compared to KVM. The HPC community has all but avoided the use of virtualization technologies due to excess impact from overheads from cpu utilization and I/O workloads. The study shows that OPENVZ is an os level virtualization which show the best solution to achieving near native performance of CPU and I/O performance which is highly recommended for HPC workloads.Regola and Ducom (2010)

2.1.3 Virtual Machine Vs Container Tools/Metrics/Resource Type

Wagner et al states the virtualization technology is an important field with advantage's such as isolation, migration, and flexibility for cloud infrastructure. The study also shows that traditional virtualization techniques are not suitable for IoT technology and embedded system technologies as they have huge performance overheads hence we need more lightweight approaches to accommodate these new futuristic technologies. Container based virtualization is a suitable alternative as we can see from the study that it provides performance to that of bare-metal and is ideal for the requirement of IoT based applications as edge devices need to respond as fast as possible to a distributed connected sensor requests. The study uses the Rasberry pi 2.0 model for the evaluation procedure considering benchmarking tools such as IOZone for disk performance and STREAM for memory performance while NBP was considered for process performance benchmarking. The experiment conducted by the study shows us that container based virutalization showed much lower performance overhead and power consumption as to that of native performance which is ideal for IoT based virtualization although some cases of overheads were noticed during the experiment it was still close to that of native which is much better than traditional virtualization approach. dos Santos Marques et al. (2018)

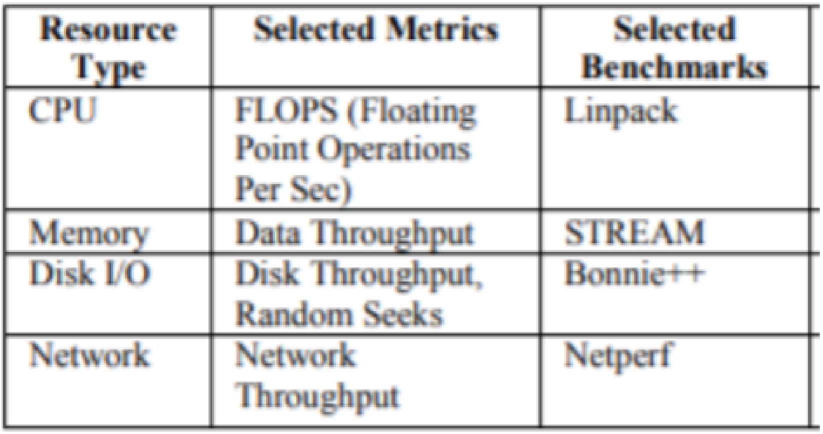
Cloud computing mainly deals with shared resources which are generally distributed to different application as per their requirement but machines requesting resources from a shared pool always tends to cause performance degradation which is an unwanted factor in any cloud service. To prevent this interference applications need to be isolated from each other to enabled isolated run-time environments for each application without them interacting or interference and the simplest way to achieve this isolation is by containerization of the application.Krebs et al. (2014)

Belson et al study of Doclite benchmarking shows us that typically benchmarking of distributed systems are done by benchmarking entire VMs utilizing all its resources even though this gives us the information to select VMs that maximize performance of the target it fails to consider the real-time use of the VMs during benchmarking. Hence the study looks at using containers based virtualization for more lightweight and real-time benchmarking as an alternative to the current benchmarking solutions. The study further divides the benchmarking process into two main division Native and Hybrid where the hybrid employs containers to benchmark cloud VMs in real-time while Hybrid looks to benchmark the entire VMs by running a benchmarking tool on an entire VM. Although this study shows Doclite to be more faster with benchmarking compared to other tools used by researchers we notice that DocLite looks to favour the performance of docker containers more than other linux based container.Varghese et al. (2016) Container have potentially been recognized as based technology for many distributed large scale systems although the technology is in its infancy there still remain many open issues with the technology in specifically of resource management. All thought there are many studies showing performance anlysis using various benchmarking tools to the best of our knowledge there are no standardized benhcmarking or monitoring tools that enable to measure the performance of workloads on containers. The main parameters of monitoring are as follows

- Resource Usage- In Cloud computing most monitoring tools that is considered to track resource usage like CPU,network and memory.

- Services - In cloud computing most monitoring tool are designed to verify if the workload is executed by the application as required.
- Status- In cloud computing the monitoring tool should help optimize resource utilization based on set optimization goals.
- execution- In Cloud the monitoring tools should track if the workload is executed without any failures in the system. This type of detection can be done by two different method which is periodically and the other being by a detection of failure of running workloads.

The growing popularity of cloud computing has made most cloud providers to adopt resource sharing strategies. Virtualization technologies like LXC containers are becoming the next generation of cloud with the rise of docker we see the new adoption in this technology for various cloud providers as the hypervisor based approach causes performance degradation with various futuristic technologies. We look at a performance study between KVM and docker container running a performance analysis considering CPU utilization and Disk I/O workloads running benchmarking tools like cAdvisor while docker stat is used for the measurement. Prometheus and Grafana are both used to enable us to query and visualize the data we require. The study uses a combination of three tools to consider the performance analysis which shows us that there are no standardized tools for benchmarking the performance analysis of Container and Virtual Machines. There is huge requirement for defined performance metrics and standardized benchmarking tools for the performance evaluation of these virtualization technologies.dos Santos Marques et al. (2018) R.Buuya et al proposed a study for performance metrics and benchmarking tools for containers and virtual machines for various parameters such as CPU utilization, Memory performance, network performance, and Disk I/O. The following figure below shows the tool considered for the following parameters to benchmark the performance analysis.



Resource Type	Selected Metrics	Selected Benchmarks
CPU	FLOPS (Floating Point Operations Per Sec)	Linpack
Memory	Data Throughput	STREAM
Disk I/O	Disk Throughput, Random Seeks	Bonnie++
Network	Network Throughput	Netperf

Figure 2: Resource Type/Metrics/Tools
dos Santos Marques et al. (2018)

2.1.4 Container Consolidation works

One of the most challenging aspect for cloud provider is to reduce the migration time and power consumption of their data centers. We have noticed a great shift in paradigm for traditional hypervisor based virtualization approach to a container based clouds which have many advantages over the traditional approach. The large part of the literature investigates resource management techniques for both level of virtualization OS and System level virtualization. Since CaaS cloud model has newly been introduced all research conducted can be grouped under OS level virtualization which increase flexibility, scalability and agility.Many major cloud providers have made use of this system enabling container based clouds by following this approach applications can be deployed on more lightweight approaches and avoid performance overhead on the other hand the hypervisor based approach allows the user to choose the OS they want although cloud is not inter-operable as such windows and linux cannot be used on the same hardware.Xavier et al. (2015) In this section we look at various means and strategies for container consolidation to improve migration time and other factors like energy consumption and performance.

Mu et al proposes Brownout approach for efficient management of resources in cloud environments although this technique shows great promise in hypervisor based virtualization the same technique is shown to have been very challenging to implement on container based environments of cloud. Although the Brownout approach is a difficult with containers but the integration of Brownout technique with containers will be a promising direction and will solve scheduling problemsXu and Buyya (2019).
Dong et al shows the placement of containers using a greedy algorithm approach by placing containers at the most efficient servers first as compared to the most required servers first. The experiment conducted shows that the resource management is better when compared to random scheduling scheme and more significant compared to least allocated server first scheme.Dong et al. (2014)
Yaqub et al provides a comparative analysis between IaaS and PaaS cloud deployments showing that containers consolidation on PaaS cloud deployment model was more on terms of OS level virtualization and a comparatively lightweight approach where the applications were deployed on containers and allocated according to the requirement and de-allocation as when there is no requirement. But this model generally shows use under utilization of resources. The contribution provided by the study involves the modeling of service container consolidation problem.Yaqub et al. (2014)
Sareh et al proposes a Container model which uses containers as the building block making it CaaS model. The study presents that re-sizing of VMs is required for the proposed consolidation technique to apply either machine consolidation or DFVS dynamic frequency and voltage scaling but we see that VM taken at a fixed size is shows better resource utilization and also VM sizing is still a very difficult technique to implement at an enterprise level.Piraghaj et al. (2015a)
Belogalazov et al proposes a technique of resource management by turning off idle nodes during the during the consolidation process in virtual machines although this seems like a good way to manage resources the start-up and shutdown time are higher on VMs as compared to container and deploying containers on underlying virtualized environment for better isolation will increase performance degradation and increase energy consumption although the study shows promising results on planet lab cloud platform it still does not take into account time consumption of start up and shutdown time.Beloglazov and Buyya (2010)
Kim et al propses a scheduling algorithm to harness the capacity of CPU using DVS algorithm is showed to be both time shared and space shared resources. The proposed algorithm is simulated and showed to be resource efficient and energy efficiency.Kim et al. (2007)
Ehsan et al has looked at reducing performance loss and resource consumption in cloud data centers by proposing a multi-criteria algorithm which looks at aspects of finding out the underloaded host and virtual machine placement. They have used a enhanced policy for resource allocation they have simulated the algorithm on cloudsim and considered resources such as cpu, disk, memory and VM migrations the algorithm shows significant improvement on energy consumption reduction on cloud data centers.Arianyan et al. (2015)
Arianyan et al has looked at multi-target resource management or allocation in the cloud environment which broadly looks at resource allocation problem. The paper looks at implementing a generic algorithm to improve the performance and reduce energy consumption. They have used cloudsim to simulate to the algorithm and the results show that the energy reduction and performance outperform that of the state of the art Arianyan et al. (2016)
Minxian et al provides a study of adopting the brownout approach in container based clouds. They have adopted the brownout system architecture by turning off optional container in applications also unused microservices by considering various policies to find the containers that can be deactivated. They look at a policy which is Minimum number of containers first which looks at provisioning only the required containers as required for incoming workloads this approach is backed by real data traces and show reduction in resource consumption while maintaining quality of service.Xu et al. (2018)
Neeraj et al has provided an study on classification of jobs based on multi-indexing and a scheduling based approach in container based clouds. They have designed a model using container clouds and showed that their model is more resource efficient in terms of host selection than the current state of the art approach.Kumar et al. (2018)
Abbas et al have looked at quality of service of virtual machine consolidation they look at an algorithm for finding under- utilized hosts and also an efficient placement of virtual machines. The study utilizes the virtual machine based dynamic threshold algorithm to find the under loaded hosts. They model and test out the alogirthm on cloudsim to simulate a real cloud environment.The simulation shows improved quality of service and better energy consumption.Horri et al. (2014)

Niloofar et al proposed a joint Virtual machine and container consolidation they also consider a new system architecture that looks at flowchat to handle to various problems of container and virtual machine migration problem. A JVCMMD policy which look at virtual machines or containers should be migrated simultaneously. The study uses CloudSim also as a simulation model to look at improved migrations per virtual machine Gholipour et al. (2020)

2.2 Conclusion

The entire review of literature has enabled to show that container based virtualization is a more lightweight approach when compared to hypervisor based virtualization. Although containers do experience some performance overheads but it shows performance to that of near native. some researchers have suggested the container could be used as the next base for building the new cloud environments Felter et al. (2015). We also look at the performance of futuristic technologies like IoT, Hadoop, and embedded workloads from the various literature we have seen that the performance overheads experienced by hypervisor are not experienced by containers they perform close to that of near native and are more lightweight approach they also provide the applications running in them with an isolate runtime environment. Containerization of applications is the new approach that provides an approach to avoid resource conflict and interactions between different applications. It also provides the applications running within them with additional advantages like an isolated run-time environment, portability, and scalability. Through the literature review we look at different experimental setups and types of containers and virtual machines to perform the benchmarking between the virtualization environments. We consider KVM for the hypervisor based virtualization for its more stable and overall and most common hypervisor based virtualization, For OS-level virtualization we choose docker container as our container of choice because it is the most common linux based container and the leading player in the market of containers. Although we notice most of the research look to also consider benchmarking network performance and memory performance we consider only CPU utilization, File I/O, and MYSQL based workload as parameters for our benchmarking. As part of the literature we look at various tools and metrics used for benchmarking the performance of either Hypervisor and container environments although we came across alot of promising tools for the benchmarking tool we choose to use sysbench as it widely used by most researchers. Sysbench is a benchmarking tool used for benchmarking system level parameters running database related workloads. Sysbench helps in can help in the experimental benchmarking of CPU utilization, File I/O and MySQL workload. We choose MYSQL as our database to stress both virtualization environments as it is the most commonly used relational database.

The vast literature review of related consolidation techniques used to improve resource management techniques during container consolidation for better migration time, SLA violations, Energy efficiency, improved performance, Improved Container placement policies, VM allocation policies. Although most of the literature looks at SLA violations and virtual machine migration problems. Our thesis looks at implementing a novel algorithm for improving the selection time of containers to be migrated to the best of our knowledge there has been no algorithm which has been implemented to improve the selection time process of container. The literature has also given us vast knowledge about CloudSim which is a simulation tool used to simulate the cloud environment. We look to simulate Container based cloud environment on Cloudsim and set up the selection time algorithm as part of the selection module policy. We look at comparing the migration time and performance in terms of resource management between container consolidation and vm consolidation which will also enable us to show that Container based clouds are the future of most cloud computing environmentsFelter et al. (2015).

3 Methodology

In order to carry out this thesis we utilize two means of mythologies for accurate evaluations of the proposed work. The first phase of this thesis looks at a performance analysis of virtual machines and Container with respect to CPU utilization, File I/O, and MySQL workloads we choose the emulate the study using a fixed physical resources and scaling up the the number of guests per host while checking the impact on the selected performance parameters. We choose Sysbench as our benchmarking tool of choice for the given experiments. The experiments are carried out 10 times to avoid any ambiguities in results. we also plot the noted values to show the comparison between each environment.

The second phase of this analysis looks at comparing the consolidation process of containers for a time selected policy of migrations of containers in CaaS clouds. As such we will not be able to emulate an entire cloud data center for our experiment or lease an entire data center due to the cost of such huge infrastructure requirements for the experiment. Instead we look at simulation as an option for conducting this evaluation. We use CloudSim as a cloud environment simulator to simulate huge infrastructure requirement's for research basis. The literature review has provided us with vast information about the use of this tool for simulation of research. In order to evaluate the proposed algorithm we use cloudsim to model the cloud environment where we run the consolidation process for number of containers migrated on a fixed interval of time. As such we need a base to compare the consolidation performance we look to compare container consolidation with that of virtual machine consolidation using the same fixed policies for both environments the values of migration time and energy consumption are noted for a number of runs and a sample T test is conducted to evaluate the results. We also use visual aids of box plots for better explanation of obtained results.

4 Design Specification

4.1 KVM Vs Docker Benchmarking

The first phase of this thesis consists of performance analysis of Virtual Machines and Containers running on the same physical host. The parameters that we consider for performance anlysis of the following virtualization technologies are as follows.

- CPU performance.
- File Input/Output performance
- MySQL server performance.

The experiment is conducted on a linux operating system. It is important for us to consider virtualization technologies that support the linux based environment we choose KVM as the hypervisor-based virtualization and Docker container as the OS-based virtualization of choice for our experiment. We choose to run MYSQL server database as workloads for our benchmarking experiment we choose MYSQL is because the database is very commonly used and can also considered to stress test VMs and Container during the experiment. System specification

- Processor-Intel (R) Core (TM) i7-8850U CPU@1.80 GHz.
- Architecture X-86-64
- RAM 8GB.
- HDD Capacity 1TB
- HOST OS LINUX 16.04

Sysbench is a benchmarking tool used for benchamrking system level parameters running database related workloads. Sysbench helps in benchmarking the following parameters database server performance, File I/O, and CPU utilization.
The experiment will enable us to stress test CPU for desgineated workloads. The second test will stress file I/O using certain operation like read and write onto the MYSQL database for both KVM and Docker virtualization environments. The experiment will be performed over 10 times to avoid any ambiguities in results. Replace this text with your Research Methods and Specifications information

4.2 Container Consolidation Model

The proposed model looks at container as a service model where application or workloads are directly executed on a containerized environment. The request for the service is generally provision on request of incoming workload or application requirements. Containers are generally run on physical servers or inside virtual machines which are dependent on CPU, memory and network performance. The main objective of this model is the consolidation of containers to improve migration time and show improved performance using a novel selection algorithm.

- **Host Overload/Under-load-** This is the component that is used to check the resource status every couple of minute's weather the host is under or over utilizing resources. The main goal is to identify under utilized hosts when they is done the host id is returned to the container module with the list of all container running on that host is consolidated.

- **Container Selector-** The main goal of this components is triggered when the host is overloaded and to provide a container to this component till the overload status of the host is no longer detected.

- **Container Migration List-** This component holds information of container that were selected by the selector component.

- **Overload Host List-** This component host data of all host that display or trigger an overload status.

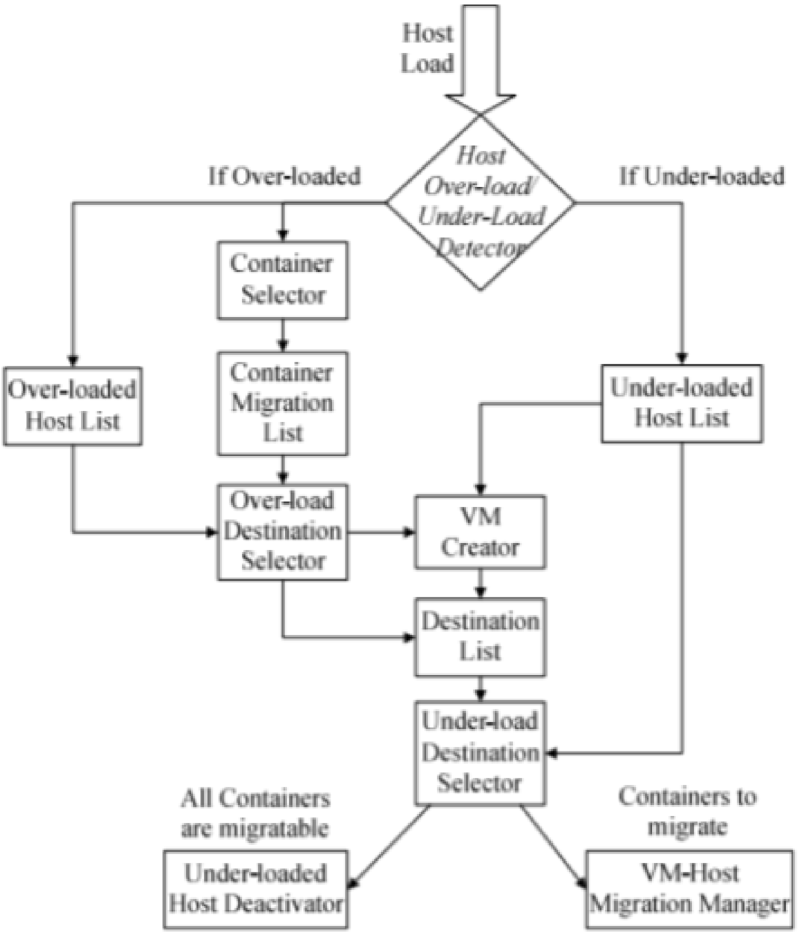


Figure 3: Container Consolidation model
Piraghaj et al. (2015b)

- **Overload Destination Selector-**This component is used to find the appropriate destination for the container to be migrated from the container list.

- **Destination List-** The migration destination is generally stored in this section along with the container and VM ID.

- **VM Host Migration Manager-** This component is used to trigger migration.

- **Under Loaded Host Deactivator**-This component is used to turn off underloaded hosts that have their containers migrated.
- **Under Loaded Host List**- This component is used to generate and find the best destination for container to be migrated from the underloaded host.

4.3 Container Consolidation Algorithm

Algorithm 1 Container threshold detection

```
Input CPU utilization of host (CUH)
Output Host status
$CUH > over - loadthreshold$
host status ← $over - load$
$CUH < under - loadthreshold$
$hoststatus \leftarrow under - loaded\ else$
$hoststatus \leftarrow Ideal$
return host
```

Algorithm 2 Container Migration time selection

```
Input SeverContainerList (SCL)
OutputSelectedContainerList
$MigrationTimeList \leftarrow CalMigrationtime$
$MigrationTimeList.sortMigrationtime()$
while host status is not ideal do
$pop()$
Return container with min migration time
$Container \leftarrow MigrationTimeList.pop$
$SelectContainerList.add(container)$
$SCL.remove(container)$
```

Algorithm 3 Container placement algorithm

```
Input Host list
Output Destination
while HostList is not empty do
$Host \leftarrow HostList.pop()$
if Host meets the container resource requirement
return Host
return null
```

- The container threshold detection algorithm is usually deployed into the host detection module. The main functionality of this algorithm is to detect if the host status is under-loaded or overloaded. We input the a signal to check the CPU utilization of the host to enable us to understand the total utilization of the host in terms of resources consumed which enables to algorithm to output the status.

- The container migration selection time enables us to choose the container from the container from the container migration list based on the migration time that is calculated. The algorithm will then pop the container with the least migration time first if overload status is detected. We input the entire container list of all container available. The alogirthm then sorts all containers in the list based on migration time and arranges them based on containers with least migration time. The sorted container migration list will have containers with the least migration time ready to migrate first as per the host requirements.

- The container placement algorithm uses the first fit approach by which it places the container on the host that first meets the container resource requirement once the container is selected from the container migration phase. The host list is fed as input the algorithm then checks if the host list

5 Implementation

5.1 Virtual Machines vs Container

The First part of analysis that is conducted in this thesis is the performance analysis between Hypervisor based virtualization and OS- based virtualization. We look at testing out CPU utilization , File I/O, and MySQL workloads on either of the environments and check the difference between both environments in terms of set parameters.

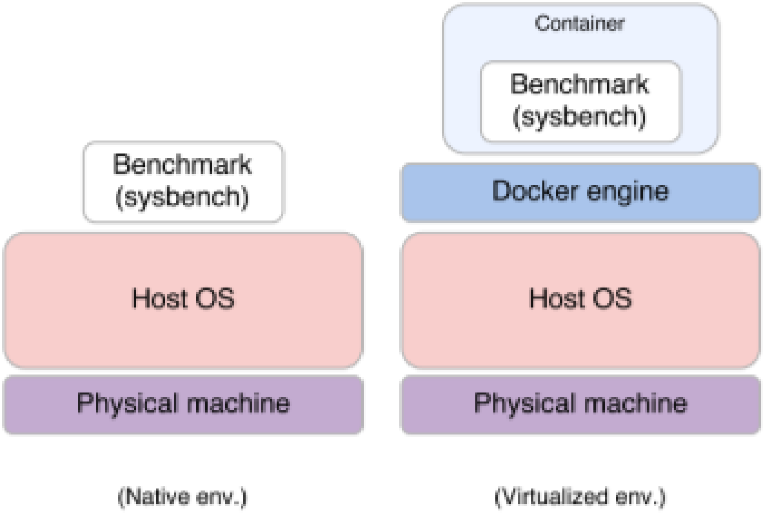


Figure 4: Benchmarking VMs vs Docker
Casalicchio and Perciballi (2017)

We consider 8 hosts of Virtual Machines and 8 Docker Container to conduct the analysis. KVM is chosen as our hypervisor of choice while docker containers are chosen as the OS-level virtualization of choice. We choose MYSQL database to perform our experiment we consider MYSQL amount the rest because it is the most commonly used database system in most environments and can be used to stress the environments for the experiment. As such we need to install MySQL-server and sysbench on all the hosts of VMs and containers to conduct the benchmarking. The experiment is repeated for about 12 times to reduce any ambiguities and the values of the benchmarking are considered for the calclulation. we look at KVM as the native

5.2 Container Consolidation

To evaluate the container consolidation we consider the Cloudsim Simulation tool to enable us to simulate the cloud environment and perform the required analysis of implementation of the container selection policy algorithm as part of this thesis. Containers have been showed to be more lightweight virtualization technology is comparison to Hypervisor based virtualization layers. The containers provide an isolation of workloads without the overheads generated by hypervisors.

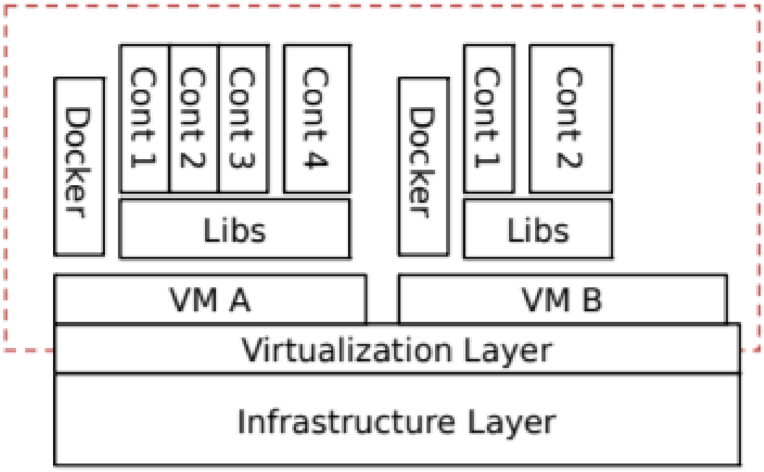


Figure 5: Container Consolidation Model In CloudSim
Piraghaj et al. (2015b)Piraghaj et al. (2017)

The virtual machines on which the containers are deployed add an extra layer of security and isolation to unknown incoming workloads. Simulation models help us to compare innovating resource management strategies the simulation model enables us to simulate large scale distributed infrastructure to perform experiments of resources management polices rather than having to physically access cloud infrastructure which becomes a very expensive and tedious task. The simulation also enables to verify and test out various resources management policies.

5.2.1 CloudSim

CloudSim is a simulation kit that enables us to simulate a cloud environment. It also provides us with a model of behavior for various cloud components such as virutal machines, data centers, and users.Containers are a new technology that is just in its infancy so we require an development environment to test various resource management polices and CloudSim provides us with such an environment to test such policies without being dependent on having a physical infrastructure during an early phase of research. Container CloudSim has layered architecture which comes along with the following features

- Workload Management Service
- Virtual Machine Life Cycle Management
- Container Life Cycle Management
- Resource Management Service
- Power and Energy Management Service

The experiment we aim to perform to improve migration time and performance improvement uses Resource Management Service of CloudSim which have several operations that can be performed under this service such as Container placement, VM placement, Consolidation, Container Allocation, VM allocation.Piraghaj et al. (2017)

5.2.2 Container Provisioning

The simulator provisions containers by the following methods at VMs and at containers. When containers are provisioned to VMs the processing power provided to each container has to be specified but at container level there can be a fixed number of resources assigned to each application running within the container. Also Container RAM and Bandwidth provisioning are abstract classes that enable the provisioning of RAM and Network bandwidth to the running containers as per required .

5.2.3 Container Allocation Policy

The main goal of the container allocation policy is to select a VM for which a container can be assigned that satisfies the resource requirement.

5.2.4 Virtual Machine Allocation Policy

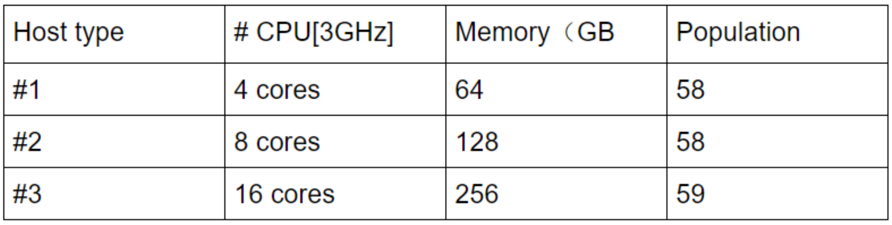
The main function of this is to allocate Virtual Machines to the host also it manages and implements allocation policies at the VM and Container for consolidation.

5.2.5 Workload Management

Most Cloud environments experience huge variable workload which is one of the main characteristic of application running on the cloud. The simulation tool CloudSim also enables us to model workloads as the same pattern of application that are running on cloud environments. We use the exisiting Utilization model in cloudsim to enable us to generate various workload patterns.

5.2.6 Container Consolidation

The aim of the study is to use CloudSim simulation to investigate the proposed algorithm to test out the efficiency during the consolidation process in terms of migration time and energy consumption in a cloud data center. To carry out this experiment the setup mimics the setup of Piraghaj et al. (2015b) We simulate a data center with about 175 physical nodes three type of 1250 containers and four types of 250 virtual machines based on Container as a service framework in container cloudsim. The characters tics of the different types of host are showing in tbe below table.



Host type	# CPU[3GHz]	Memory（GB	Population
#1	4 cores	64	58
#2	8 cores	128	58
#3	16 cores	256	59

Figure 6: Types of Host Configurations

A data center with 175 heterogeneous servers of three different types is simulated. Characteristics of each server together with VM and container configurations . Network bandwidth is 1 GB/s, 10 MB/s, and 250 KB/s for servers, VMs, and containers, respectively. The same assumption is made for disk bandwidth and it is 1 TB, 2.5 GB, and 0.1 GB for servers, VMs, and containers, respectively.

6 Evaluation

6.0.1 KVM vs Docker CPU utilization Benchmarking

In the first section we look at benchmarking the CPU utilization of Hypervisor based virtualization and OS-level based virtualization. We look at KVM as our hypervisor based virtualization of choice and Docker container of choice we look at running 8 VMs and 8 Containers for our analysis. CPU performance depends on the following factors time required to execute a particular task and the maximum and minimum time required to process a single request.

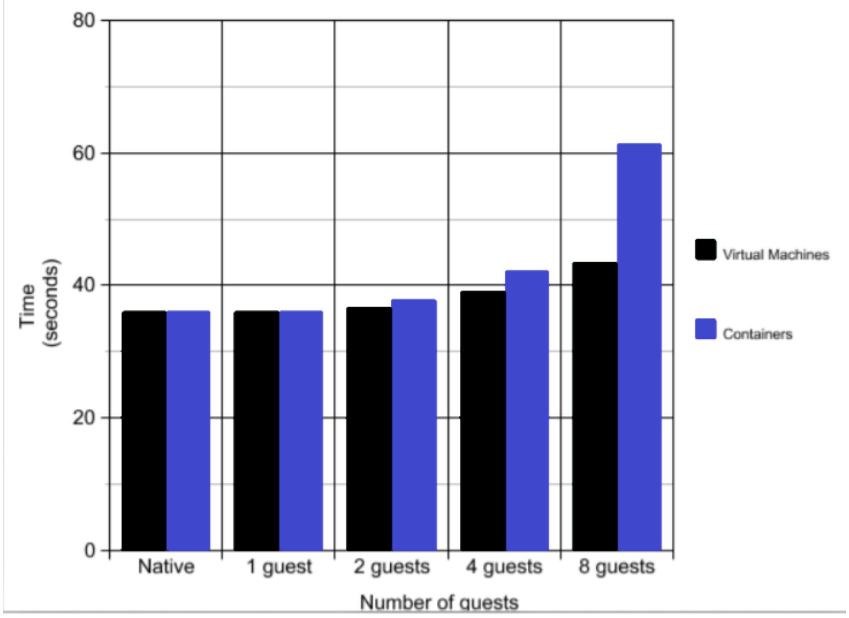


Figure 7: Container Vs Virtual Machine CPU Benchmark

We start the process by benchmarking the native performance and then we conduct the sysbench benchmarking test running the command ”sysbench –test=cpu –cpu-max-prime=30000 run ” for 30000 prime numbers on each host as we systemically increase the number of guests per host. We calculate the total execution time.
As we see from the above figure we notice that the time required for a process to complete a given task in VMs is not that high when compared to containers but as we increase the number of guests to a given host we notice slight delay in terms of performance in virtual machines compared to that of container although the difference is not big enough to make much significant difference but when we are limited to a fixed number of physical resource for our experiment it would be intersting to see how the performance degradation of CPU perfoamnce would be at scale of spinning up about 1000 VMs to that of Containers. Although for our current analysis we would suggest that there is no major difference with choosing one environment to the other and should be chosen mainly on the functional requirements.

6.0.2 KVM vs Docker FILE I/O

The way in which we need to stress the test each environment for input output is by preparing a test file which is more than that of the RAM size so that the system does not use the RAM for caching. We prepare a test file of 80GB using sysbench ”sysbench –test=fileio –file-total-size=80G prepare” this will enable us to prepare a test file for the experiment. We consider the total read/write operations performed in each environment as the parameter for our analysis. we use the sysbench tool to conduct the read and write operations using the command ”sysbench –test=fileio –file-total-size=80G –file-test-mode=rndrw –init-rng=on –max-time=300 –max-requests=0 run”

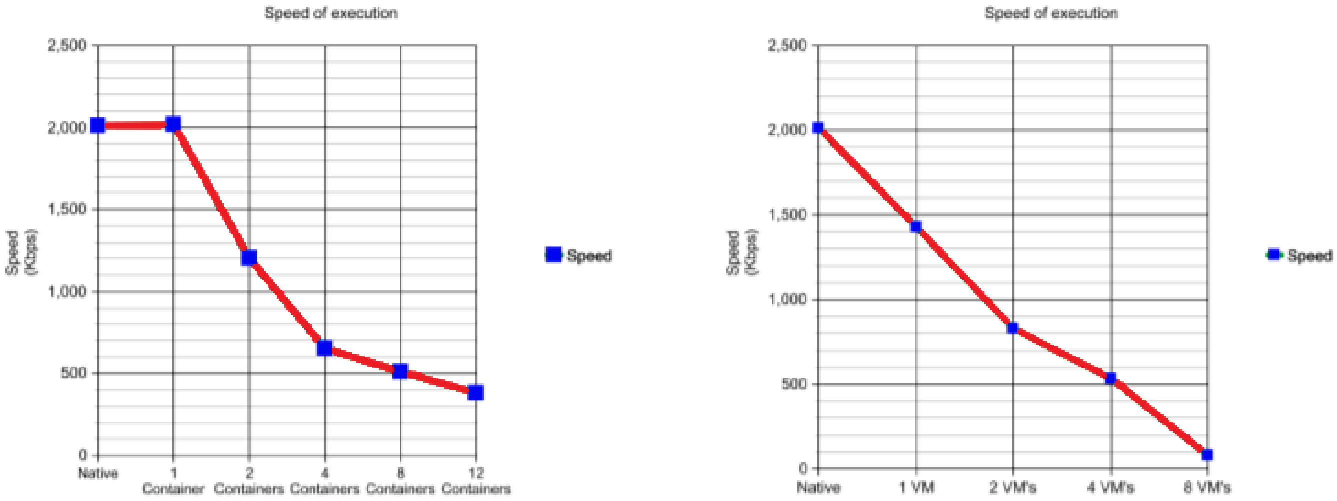


Figure 8: Container Vs Virtual Machine File I/O Benchmark

The most important parameter for our analysis is the speed or speed of execution of file I/O in either environment as we from the above graph the speed is starts of 2Mbps. The above graph for virtual machines shows us that there is a steep degradation in performance which is directly linked with the number of guests increased on the host. While containers we see the performance degradation stable at a point which is a huge difference when compared to virtual machines hence Docker outperforms KVM in this case the reason for the difference can be seen beacuse of the feature of dynamic allocation of resources in container unlike virtual machines where resource allocation is fixed.

6.0.3 KVM vs Docker MySQL workload

To benchmark the database on each environment we first install MySQL in all hosts and then setup a test database with 1,000,000 rows of data using sysbecnh we can prepare the table "sysbench –test=oltp –oltp-table-size=1000000 –db-driver=mysql –mysql-db=test –mysql-user=root –mysql-password=yourrootsqlpassword prepare" once the table is prepared we can run the sysbench test and systematically increase the number of guests per host. We run the "sysbench –test=oltp –oltp-table-size=1000000 –db-driver=mysql –mysql-db=test –mysql-user=root –mysql-password=yourrootsqlpassword –max-time=60 –oltp-read-only=on –max-requests=0 –num-threads=8 run" we consider the parameter of total number of events executed in each environment as we systematically increment the host vs guest.

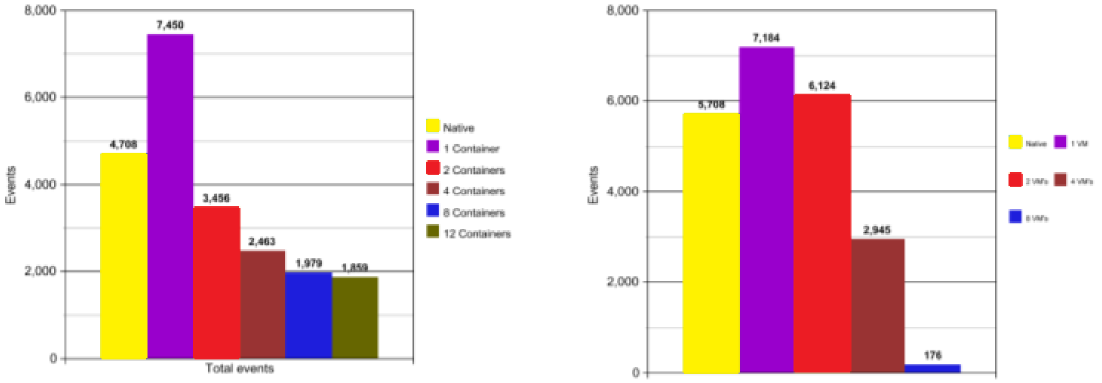


Figure 9: Container Vs Virtual Machine MySQL workload

As fair analysis of performance we choose to analyse the total number of events executed under each environment from the above table we can notice that the events start of the same but as we scale both environments to about 8 we see that virtual machines have about 176 events running on them while containers have 1979 events running on them this shows us that with scaling up the number of resources containers till have much better performance as compared to virtual machines. Although we could look at the total number of read , write operations performed we feel the events executed in each environment holds a better parameter for fair analysis of performance. Thus we can state that mysql server runs or performs better in docker containers than KVM.

6.0.4 Virtual Machine Vs Container Consolidation

In order to check the performance of container consolidation we set virtual machine consolidation as the benchmark to check the performance of the consolidation we consider paramters such as migration time, number of container migrated and also with the help of CloudSim we can calculate the energy consumption of containers to that of virtual machines. We run the consolidation simulation of both containers and virtual machines using the same allocation policy. we use the first fit policy of host selection in both cases of the experiment. We simulate a data center with 175 heterogeneous servers of three different types is simulated. Characteristics of each server together with VM and container configurations .The same assumption is made for disk bandwidth and it is 1 TB, 2.5 GB, and 0.1 GB for servers, VMs, and containers, respectively. The experiment is performed 10 times to avoid any ambiguities if results and the values are box plotted against each other.

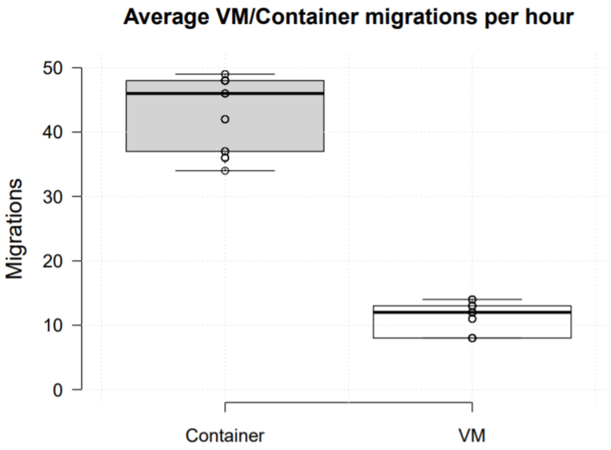


Figure 10: Container Vs Virtual Machine Number of Migrations

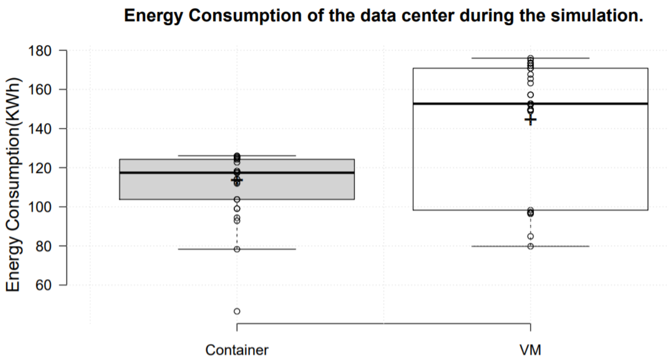


Figure 11: Container Vs Virtual Machine Energy Consumption

We conducted a sample T test from the values we received and notice that the number of containers that are migrated per virtual machine is 531 and energy consumed is 49.088(KWH) more than the number of Virtual machines migrated which is 209 while power consumption is around 113.92 (KWH). The values are taken for long cycles of runtime and the graph is plotted based onn values obtained from the simulation. The above graph shows us the overall advantage of container consolidation as we test it against Virtual machine consolidation using the same polices of selection. We see that the Container consolidation being a more lightweight approach to machine virtualization not only outperforms its counter part VM but also is more energy efficient in terms of resource management.

7 Conclusion and Future Work

Virtual Machines and Containers are both promising virtualization approaches based on requirements we can choose one environment or the other although container do not allow the user to choose an operating system they are comfortable with when Virtual machines and Containers are set for better performance operations containers outperform VMs in all cases with their performance to that of near native. The experiments shows us that with scaling up the number of VMs and Containers on each host CPU remains almost similar between KVM and Docker but after a point the performance of Docker starts to get much better in comparison. While the File I/O we notice that container out perform VMs in this aspect although File I/O workloads should be selected carefully. In terms of MySQL operations on each environment we notice that again container outperforms KVM.

To check the performance and efficiency of container consolidation using the given policy we compare the same implementation running the same test on virtual machine testing out VM consolidation as the base for our comparison. The algorithm implemented shows to outperform the studied state of the art in terms of number of migrations and energy consumption. The experiment has shown us that given the choice of migration of containers to that of consolidating idle VMs in cloud environments migration of container seems to be the better option with regard to resource management. As such we have successfully achieved all objectives stated in section 1.

7.1 Future Directions

The future direction of research is still a vast scope in container although serverless computing is the next paradigm in computing environment there still remain many use cases where servelerss or FaaS-Function As A Service environments cannot match containers also we should enable the effective placement of container just like VMs which could possibly improve resource utilization in cloud data centers. As such we can also look at overbooking container as container virtualization has a feature of multi-tendency where by different workloads can co-exist on a single container although the container still runs a single application as a process but with the right placement we can run different workloads although this will cause performance interference but at an optimum setting this could enable cloud provider to overbook container and enable them to further improve resource management policies. We can also look at policies for container overbooking to check resource utilization in container which could enable us to add knowledge about co-existing workloads this study should provide us to setup an overbooking factor to enable us to set up advance overbooking algorithm for container which will enable cloud providers to overbook containers which will improve resource management to a greater factor than the current state of the art.

8 Acknowledgements

I Use this opportunity to express my deep and extreme gratitude to my mentor and infrastructure professor Mr.Vikas Sahni Professor of Cloud Computing at the National College of Ireland who's support and motivation has been a driving force for me to complete this research. The suggestions and guidelines provided by him have been very valueable for me in successfully completing this project.

I would also like to thank Dr.Horacio Gonzalez-Valez who has been a key inspiration for me to pursue my Master In Computing and also for his feedback in the initial stages of research. Finally i would like to thank my parents for their continuous. support and motivation.

References

Arianyan, E., Taheri, H. and Sharifian, S. (2015). Novel energy and sla efficient resource management heuristics for consolidation of virtual machines in cloud data centers, *Computers & Electrical Engineering* **47**: 222–240.

Arianyan, E., Taheri, H. and Sharifian, S. (2016). Multi target dynamic vm consolidation in cloud data centers using genetic algorithm., *J. Inf. Sci. Eng.* **32**(6): 1575–1593.

Beloglazov, A. and Buyya, R. (2010). Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers., *MGC@ Middleware* **4**.

Casalicchio, E. and Perciballi, V. (2017). Measuring docker performance: What a mess!!!, *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, pp. 11–16.

Dong, Z., Zhuang, W. and Rojas-Cessa, R. (2014). Energy-aware scheduling schemes for cloud data centers on google trace data, *2014 IEEE Online Conference on Green Communications (OnlineGreenComm)*, IEEE, pp. 1–6.

dos Santos Marques, W., de Souza, P. S. S., Rossi, F. D., da Cunha Rodrigues, G., Calheiros, R. N., da Silva Conterato, M. and Ferreto, T. C. (2018). Evaluating container-based virtualization overhead on the general-purpose iot platform, *2018 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, pp. 00008–00013.

Felter, W., Ferreira, A., Rajamony, R. and Rubio, J. (2015). An updated performance comparison of virtual machines and linux containers, *2015 IEEE international symposium on performance analysis of systems and software (ISPASS)*, IEEE, pp. 171–172.

Gholipour, N., Arianyan, E. and Buyya, R. (2020). A novel energy-aware resource management technique using joint vm and container consolidation approach for green computing in cloud data centers, *Simulation Modelling Practice and Theory* p. 102127.

Gill, S. S. and Buyya, R. (2018). A taxonomy and future directions for sustainable cloud computing: 360 degree view, *ACM Computing Surveys (CSUR)* **51**(5): 1–33.

Horri, A., Mozafari, M. S. and Dastghaibyfard, G. (2014). Novel resource allocation algorithms to performance and energy efficiency in cloud computing, *The Journal of Supercomputing* **69**(3): 1445–1461.

Kim, K. H., Buyya, R. and Kim, J. (2007). Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters, *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*, IEEE, pp. 541–548.

Kozhirbayev, Z. and Sinnott, R. O. (2017). A performance comparison of container-based technologies for the cloud, *Future Generation Computer Systems* **68**: 175–182.

Krebs, R., Momm, C. and Kounev, S. (2014). Metrics and techniques for quantifying performance isolation in cloud environments, *Science of Computer Programming* **90**: 116–134.

Kumar, N., Aujla, G. S., Garg, S., Kaur, K., Ranjan, R. and Garg, S. K. (2018). Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers, *IEEE Transactions on Industrial Informatics* **15**(5): 2947–2957.

Li, Z., Kihl, M., Lu, Q. and Andersson, J. A. (2017). Performance overhead comparison between hypervisor and container based virtualization, *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, pp. 955–962.

Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N. and Buyya, R. (2015a). Efficient virtual machine sizing for hosting containers as a service (services 2015), *2015 IEEE World Congress on Services*, IEEE, pp. 31–38.

Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N. and Buyya, R. (2015b). A framework and algorithm for energy efficient container consolidation in cloud data centers, *2015 IEEE International Conference on Data Science and Data Intensive Systems*, IEEE, pp. 368–375.

Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N. and Buyya, R. (2017). Containercloudsim: An environment for modeling and simulation of containers in cloud data centers, *Software: Practice and Experience* **47**(4): 505–521.

Regola, N. and Ducom, J.-C. (2010). Recommendations for virtualization technologies in high performance computing, *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, IEEE, pp. 409–416.

Varghese, B., Subba, L. T., Thai, L. and Barker, A. (2016). Doclite: A docker-based lightweight cloud benchmarking tool, *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, IEEE, pp. 213–222.

Xavier, M. G., De Oliveira, I. C., Rossi, F. D., Dos Passos, R. D., Matteussi, K. J. and De Rose, C. A. (2015). A performance isolation analysis of disk-intensive workloads on container-based clouds, *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, IEEE, pp. 253–260.

Xie, X.-L., Wang, P. and Wang, Q. (2018). The performance comparison of native and containers for the cloud, *2018 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, IEEE, pp. 378–381.

Xu, M. and Buyya, R. (2019). Brownout approach for adaptive management of resources and applications in cloud computing systems: A taxonomy and future directions, *ACM Computing Surveys (CSUR)* **52**(1): 1–27.

Xu, M., Toosi, A. N. and Buyya, R. (2018). Ibrownout: an integrated approach for managing energy and brownout in container-based clouds, *IEEE Transactions on Sustainable Computing* **4**(1): 53–66.

Yaqub, E., Yahyapour, R., Wieder, P., Jehangiri, A. I., Lu, K. and Kotsokalis, C. (2014). Metaheuristics-based planning and optimization for sla-aware resource management in paas clouds, *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, IEEE, pp. 288–297.