ISSN : 2618-1630

Optimize Elasticity in Cloud Computing using Container Based Virtualization

Noor e Sahir[1], Muhammad Amir Shahzad[2], Muhammad Sohaib Aslam[2], Waseem Sajjad[2], Muhammad Imran[2]

1: Department of Computer Science, GCU Lahore.

2: Department of Computer Science, GCU Faisalabad.

*Correspondence | Noor e Sahir **E-mail**: noorsahir23@gmail.com

Citation | Sehir e N, Shehzad M.A, Aslam M.S, Sajid W and Imran M, "Optimize Elasticity in Cloud Computing using Container Based Virtualization". International Journal of Innovations in Science and Technology, Vol 02 Issue 01: pp 01-16.

DOI |https://doi.org/10.33411/IJIST/2020020101

Received | December 01, 2019; Revised | December 26, 2019; Accepted | December 28, 2019; Published | January 01, 2020.

Abstract:

Cloud computing emphasis on using and underlying infrastructure in a much efficient way. That's why it is gaining immense importance in today's industry. Like every other field, cloud computing also has some key feature for estimating the standard of working of every cloud provider. Elasticity is one of these key features. The term elasticity in cloud computing is directly related to response time (a server takes towards user request during resource providing and de-providing. With increase in demand and a huge shift of industry towards cloud, the problem of handling user requests also arisen. For a long time, the concept of virtualization held industry with all its merits and demerits to handle multiple requests over cloud. Biggest disadvantage of virtualization shown heavy load on underlying kernel or server but from past some decades an alternative technology emerges and get popular in a short time due to great efficiency known as containerization. In this paper we will discuss about elasticity in cloud, working of containers to see how it can help to improve elasticity in cloud for this will using some tools for analyzing two technologies i.e. virtualization and containerization. We will observe whether containers show less response time than virtual machine. If yes that's mean elasticity can be improved in cloud on larger scale which may improve cloud efficiency to a large extent and will make cloud more eye catching.

Keywords: Grid computing, Elasticity, Virtualization, Containerization, Docker

Image

Introduction:

Cloud computing is getting immense importance due to its demand and availability of resources. Elasticity is the degree on which cloud paradigm is judged from other earlier approaches like grid computing. With the advent of time cloud computing has solved physical and maintenance cost of physical systems. It has many key characteristics like multi-tenancy, reliability and rapid elasticity. Elasticity is one of the main features defined from cloud service providers to compare cloud services. Elasticity of a cloud computing system is referred as its "ability to contract overtime over user demand". [1] Cloud works available in three service models IaaS, PaaS and SaaS. Platform as a service provides vendors to develop their services online. Software as a service provides users online services. Infrastructure as a service provides online API's and works on the physical layer of networks like physical resources, their location, and data distribution over cloud, backup, scaling and security. Here virtualization is used for managing requests according to demand with host OS. Both hardware and software-based virtualization exist. Hypervisor runs virtual machines and it pools services according to the customers demand. [2] Cloud computing has overheads and constrains over flexibility and scalability especially when diverse users with different needs wish to use cloud resources. To meet up such needs an alternative to virtualization is gaining importance especially in micro-hosting services is container-based solutions. [3] This enables bundling of applications and data in manner to deploy applications easily and best utilization of available resources. Along with solving many problems like dependency it also helps to improve response time in cloud that certainly helps to lighten the servers and decrease response time as it removes extra layer of host machine.

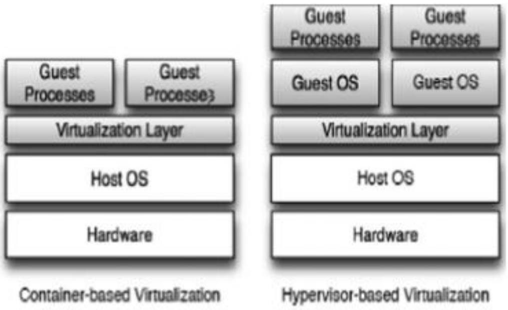


Figure1: Hypervisor Vs Containerization

To optimize elasticity different approaches have been introduced. Previous researched shows that technologies which use hypervisor face high performance overheads that lessens performance Therefore a lighter approach being introduced that is container based solution especially for micro hosting services. As in hypervisor there is always an extra layer on top of the host OS that creates an extra burden while resource provisioning and de-provisioning. While containers don't create any host layer but works in same OS and manage resources that enhance elasticity of system. With the fame of cloud computing many IT service companies start shifting their services over cloud. This is a big achievement in cloud technology but it also brings a large number of users that leads to slowing down response time. But if there is latency in services that user demand then clouds are of no use. As different users comes with different needs and wants a rapid response. We should eradicate these hurdles. One of these research paradigms is shifting services from hypervisor to container [4]. Container is an approach that completely demolishes an extra layer between core OS whereas user service demand in virtualization. There are many techniques being adopted in containerization like LXC, rkt, solarise etc. In this research we will deal with Dockers. Dockers is an open source container engine that works in many other products. It is the latest and most powerful container technology. Because it can work with older servers and also can work with ship programs. Basically, Dockers has a container HUB works like container repository. [5] Virtualization is creating virtual version of something i.e. a server, network or storage devices. Infect it is a method for using share resources by many companies and organization that are geographically apart.

Elasticity:

Elasticity is one of main cloud performance measuring parameter. From other sciences it has many definitions but in cloud it is the ability of a cloud to give sharp response to consumers according to their need and demand. It is actually measuring of time that takes by servers during scaling up and scaling down in provisioning and de-provision of resources when demand of user changes [6]. Cloud computing has become a trademark for "on-demand" service providing as it has ability to change resource "on the fly" by adding and removing them for handling load. It works on "pay as you go" model rather you are using its infrastructure, platform or application. [7] Elasticity is a model to check QoS quality of service in clouds. Cloud providers are free to add or remove updates in their projects without any interruption on the fly. Available resources for cloud users are unlimited and can purchase according to demand and need at anytime. Normally elasticity of a system is associated with scalability but there is a major difference between them. Scalability is associated with adding or removing resources according to the variation of load. While elasticity is the ability of a system to give response to consumer demands when resources are added or removed. [8] Scalability is a time free notation while elasticity is time dependent. Elasticity is the other name of system performance or matching its level of performance after provisioning and de-provisioning of resources which is a fundamental cloud principle. Hence In this thesis will discuss how important elasticity of the system is, what are propose of elasticity improvement and how Docker container can help for improving elasticity.[9] There are five main layers of network cloud which can reside which include infrastructure, kernel, hardware, application and environment. [10] It is very important to take an estimation of an average response time. In physics it is defined as the physical property of a system to retain its original position after some stress applied and response time against user request [11]. Here in cloud we term stress as a load of network and strain is bearing ability or tackling [12].$^{stress}/_{strain}$In terms of cloud it is defined as

$$^{DemandedComputingcapacity}/_{AllocatedComputingCapacity}$$

Cloud computing demand is measured in terms of GB this data is measured in terms of memory, processing time and storage ability [13].

$$cc = cpuunit * numberofcores * processingspeed * processingtime + allocatedmemory + allocatedmemory.$$

Change in it is calculated and measured in terms of resource allocation and de allocation in view of system load and bandwidth.

$$AverageBandwidth = ^{(IncomingBandwidth + OutgoingBandwidth)}/_{2}$$

Another definition of elasticity in cloud computing is refer as “configurability and expandability of resources” it means along with network bandwidth or system how resource can migrate or shift during a single request handling, which is also very important. In other words, physical resources affect the elasticity. Therefore, we focus on hardware virtualization, one main technology was VM but with passage of time and its comparatively low response, time a new technology is introduced i.e. containerization as an alternative to virtual machines.

Virtualization:

Virtualization plays a vital role in cloud computing. Normally users of cloud share application over clouds but in actual they are sharing infrastructure [14]. Through IT companies have reduced their cost to a large extent for example if you need to update a software you don't need to change it completely at every machine you just have to release updated version over cloud and it reaches all over. [15] There are four types of virtualization including hardware virtualization operating system virtualization, and server virtualization, this paper is related to compare elasticity using container virtualization which is a type of hardware virtualization.

Containerization:

Containerization is a method for packaging an application that it can run in an isolated environment with its own dependencies. Many cloud provider companies use container with its software choices like rkt, Dockers and Kubernetes. The name of container is derived from shipping industry, in spite of shipping. Each good individually a container ship different thing in a package. So container is a point that can take goods in form of one unit. Similarly all applications of one process work together in a container those are independent to others. [16] Mostly service providers face the problem of underlying hardware resource in differences. An application runs perfectly on one computer but get messy whenever shifted or migrated to other structures. This issue mostly arises when shifting applications from one server or data centre to other. This is due to the difference between machine environment, underlying libraries, storage medium, security and network topologies.[17] By using container technology this issue does not exist more as it works like a crane that picks up all shipment as one unit and place them onto vehicles for transport. Container technology carries not only software but also all libraries, binaries and configuration files. It also helps to deploy software in a server.

How containers work:

Container is not a new paradigm in industry, rooted in LINUX long ago. Recent form of container is advanced version of that technology it is quite easy to use for general purpose for deploying applications and upgrading software over cloud. [18] They also provide functionality of dividing big programs into smaller services known as micro-services. Different containers work for these parts of services interact timely and as a whole give result [19]. For faster and automate deployment in portable containers an open source project introduced Docker. Docker uses LXC with API of kernel and application level. Both of them together run applications in an isolated environment using their own CPU, memory, I/O, networks etc. Namespaces are used for differentiation of process's ID, network, process tree, file system and network [20]. Docker are created by using base images of OS. That includes basic ongoing infrastructure and repositories in a sophisticated environment. During the building often image in container every command takes a new action and creates a new layer over the previous one. These commands can execute manually or docker files use for automatic execution. [21] Every docker file is scripted having many commands to take actions against every listed command on a base image and creates a new image. These commands and images have full record of processes and provide every necessary action for deploying applications from start to end. Container gives end users an abstraction layer which makes each unit to work separately but in a collaboration manner. [22] Each process which migrates from one machine to other shifted along with all its own service routine on container, have separate process ID, network routine and other API etc. [23]

Methodology:

This research is based on reducing service response time during resource provision and de-provision. Through this the main hurdle of network load can be improved. For this we take hardware virtualization and judged its functionality thoroughly we observe that workload on servers got affected due to limitations of virtual machines. Those limitations can be improved by using a technology 'docker'. By using this technology, we will improve the service response time toward user's requests. That in turn help to solve biggest issue of cloud technology is facing i.e. handling user requests.

From previous discussion we see in detail that elasticity is main metrics for measuring cloud QoS. There are many methods of improving elasticity, One of them is lighting "Overlying OS"S so that response time can be lessened. Then we see different types of virtualization and how Container based technology improved the way of virtualization. In this research we will compare Virtual machines and Docker-container technology.

This paper contains two main parts one is about VM virtualization second is container-based virtualization. We simulate the system by using two main software including. "Azure Microsoft" and "Desktop Docker" along "data Dog". Their configuration gives results and then compares results through graphs of CPU, I/O and memory utilization analysis.

Data Dog (Docker-Container)

Data Dog is an online monitoring tool for measuring CPU utilization, Network traffic, resource allocation and their usage which covers SaaS based cloud applications. It works for docker installed on native machine by creating an agent that resides over the system and connect it with data dog server. Working with docker is very easy. You just need to give command and it automatically pulls that specific software or services over your system in an image form. That are a big advantage as it is very light to have an image rather than that full software. It saves memory, and maintaining cost.

Docker Desktop:

This version of docker is installed from docker Hub by creating an account. This is specifically for window 10 or higher version. It help developers to create images for creating lightweight virtual machines "container" that have a process and all of its necessary dependencies. Docker follow client-server mechanism and a remote API for creating images and run applications over it [29]. Docker container relationship is same like object and class. Docker have following instances in it.

Docker Images: It is like an empty vessel which provides complete environment for creating Docker containers in it.

Docker Container: A virtual machine instance flourishing according to the instruction created in Docker image.

Docker Client: Applications or other users that take advantage of container over the cloud and communicate with docker.

Docker Host: **V**irtual machine that is running in container for handling and managing API's.
Docker Machine: A manager that mange docker hosts rather running on localhost or far apart over cloud.

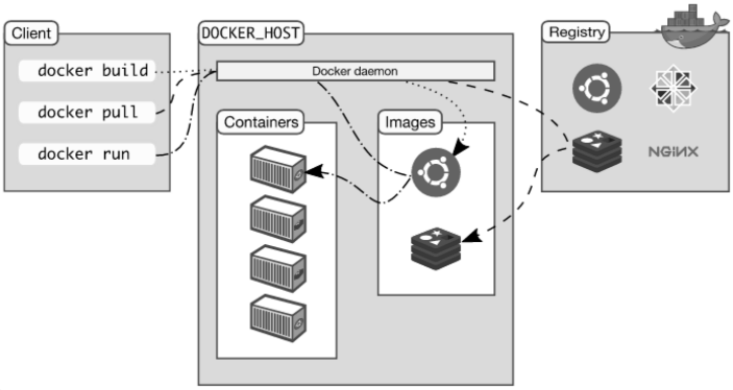


Figure 2: Docker-Container structure

Microsoft_AZURE (VM-virtualization)

It is second part of our research i.e. virtual machine based virtualization. In this we used online Microsoft Windows Azure SaaS based platform. We created an online virtual machine then linked it with cloud platform provided by Azure. Then created a dashboard for observing its working and different metrics. Here was whole process of creating VM and linked it with cloud and system. This observation was for same time as by data Dog i.e. 14 days. First we created an online account for getting Azure services then choose windows VM creation tab and started building VM. its specifications and checked simulation results after some time.

Results and Discussion:

By using both of these technology docker-container for containerization and Microsoft-azure for windows we will compare results in terms of response time, I/O network, memory, latency.

Memory:

A fix memory was assigned in our experiment and created a specific 17GB fix memory in machine. It was mounted means whereas no other services can be used in this area was spite of fact it is free or in use. That's why system having virtual machine needs more time to load. On the other hand containers don't need a fix amount of memory. It just reside in specific require memory along with all their dependencies and ship accordingly. This is why containers are considered as light weight processes and have a big advantage over virtual machines. In our experiment it just uses 52MB of memory.

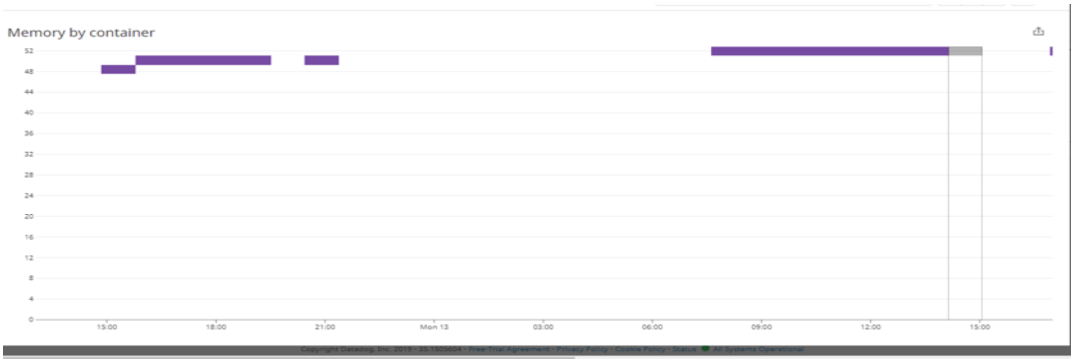
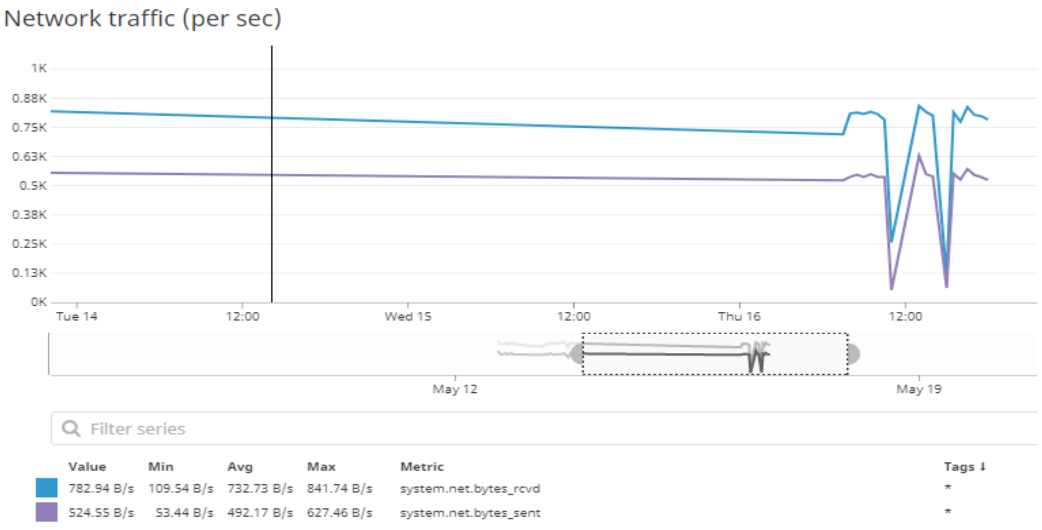


Figure 3: Container Memory Utilization

Network:

System I/O is also very important as it monitors the total load variations in the system. It is divided into two main parts to input traffic i.e. user request and system output in the form of user response. As our research work is done on personal computer that's why traffic load was not so high. Following are use case scenario showings an average overview of system traffic bytes sent and received.



	Value	Min	Avg	Max	Metric	Tags ↓
🟦	782.94 B/s	109.54 B/s	732.73 B/s	841.74 B/s	system.net.bytes_rcvd	*
🟪	524.55 B/s	53.44 B/s	492.17 B/s	627.46 B/s	system.net.bytes_sent	*

Figure 4: Network Load

CPU Usage:

CPU usage in a system is considered as the trademark for how well the system is showing benefit of processor, if CPU is idle most of the time that means system efficiency is poor and vice versa. The term CPU percentage directly relate the current working of the processor. For example, if the CPU usage percentage is 100% it means system is working at its best in that current working time period. But yes the thought of diversity of workload can't be ignored in it. So, maximum CPU usage is a sign of a good system working. In this scenario, performance of container is better on average than that of VM based virtualization. It can be observed by both of these graphs provided. On average trail system is working better over a period of 14 days. This observation was for one-hour network traffic and workload was also low. Table 1 shows that Dog CPU-usage percentage varies from 0.53% to 31.6% and on average it was 6.24%. On the other hand VM-virtualization wasted a lot of time in context switching from host to native server on average that is observed in my created environment it is 4.72%on average.

Table 1: CPU Usage

Number	Technology	Avg percentage	Min percentage	Max percentage
1	Containerization	6.24%	.53%	31.6%
2	Virtualization	4.72%	.37%	27.54%

Following are the relative graphs shown in result of simulation. Fig 5 shows results of Azure-Microsoft dashboard. In which x-axis shows time duration and y-axis percentage of CPU utilization over a specific time period.

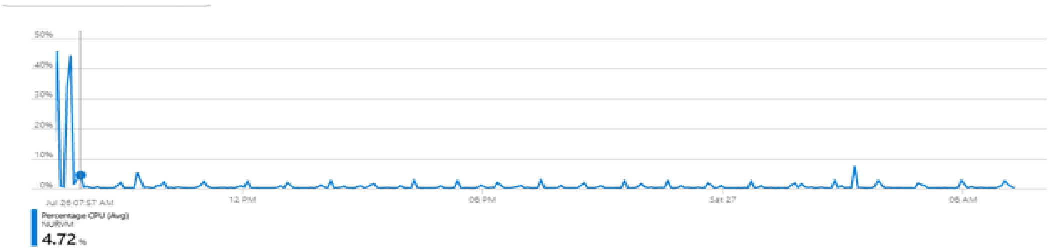


Figure 5: VM_CPU Usage

Fig 6 shows results of container based experiment on average usage of CPU. This result was taken by data dog dashboard which we set for container work monitoring in our system by creating an agent. This graph has time along x-axis and percentage on y-axis.

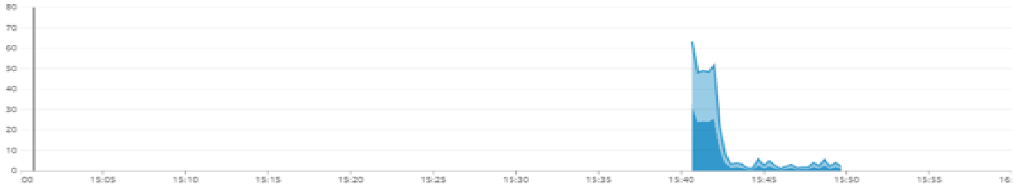


Figure 6: Container_CPU usage

An ideal system used maximum resources and from table and graph it is clear that containers utilize CPU as other part of system also use resources in spite of that idle one situation happens in case of virtual machines.

System Latency:

In cloud computing latency means the delay of response between service provider and client request. In other words it means how much a system is efficient to deal with system load during resource providing. It directly relates to elasticity of a system as main topic of our research. Latency of a system depicts the need of vertical or horizontal scaling according to the need and demand of user requests. [26] Elasticity is directly proportional to the latency. A system is more elastic if it can handle requests rapidly. We observed in detail how docker reduces time as it removes an abstract layer of virtual machine. That is an extra overhead and load to native kernel machine. It also removes many dependencies.[33] It can be seen by following results shown in Table 2 latency change from 20msto 56ms, while at docker it varies from 10ms to 44ms.

Table 2: System Latency

Number	Technology	Avg Time	Min Time	Max Time
1	Containerization	44ms	15ms	49ms
2	Virtualization	48ms	20ms	56ms

Finally, it is clear that docker based virtualization is more elastic than that of VM. Difference is smaller as it has less network flow on PC, whereby large-scale difference is quite impressing. Following are graphs that shows result of dashboard of both Microsoft-Azure and datadog dashboard having time duration of observation along x-axis and response time of system along y-axis.

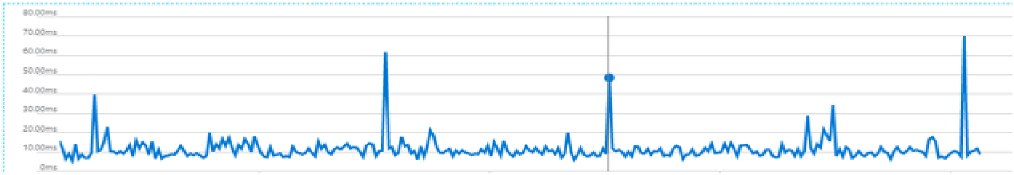


Figure 7: VM- System Latency

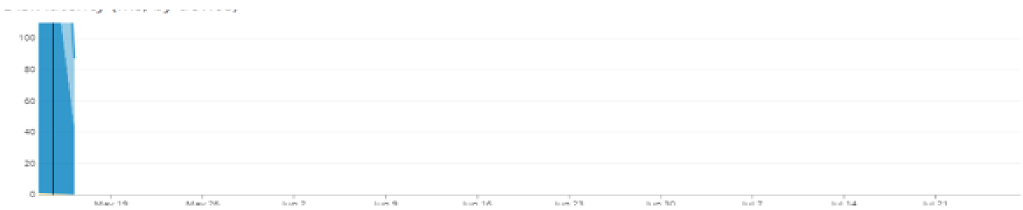


Figure 8: Container-System Latency

From Table 2 and Fig 7, 8 it is clear that response time of container based technology is short as a result elasticity of system improves.

Conclusion:

As we discussed, that the worth of a cloud infrastructure elasticity is very much important. There are many methods to improve elasticity one of them is hardware virtualization. Traditionally, it has only one method that was to create virtual machines over host machine but from some past decades a new technology gained much importance i.e., containerization etc. The main point of our thesis was to prove that using docker technology we can enhance elasticity of a cloud. Docker is a tool for creating containers in host server that diminishes an extra layer in server during resource provisioning and de-provisioning. It improves response time of system. Elasticity is main measure in cloud industry. You can refer it as a key point for checking integrity of a system. In our research we proved that a new technology that is flourishing so rapidly directly affects elasticity. Whenever response time improves, users request fulfilled faster. For this we use docker technology which doesn't create an extra layer over kernel and works like in a same OS which removes many dependencies and overhead. Which increases the response time? In future, we hope to apply docker technology for more improvement in cloud infrastructure, and security improvement which is still an open question in industry. Although container uses a whole separate group, for handling processes but there are some open issues still need to be discussed. Moreover, we hope to check more parameters on the basis of docker container for cloud improvement when system load change on a large scale, I/O control etc.

Acknowledgement: We all acknowledge our parents for supporting us throughout.

Author's Contribution: All authors contributed equally in this research.

Conflict **of interest:** We declare no conflict of interest for publishing this manuscript in IJIST.

Reference:

1. Al-Dhuraibi, Y., Zalila, F., Djarallah, N., & Merle, P. (2019). Model-driven elasticity management with OCCI. IEEE Transactions on Cloud Computing.Foster, I., Zhao, Y., Raicu, I. and Lu, S., 2008. Cloud computing and grid computing 360-degree compared. arXiv preprint arXiv:0901.0131.
2. Li, B., Gillam, L., & O'Loughlin, J. (2010). Towards application-specific service level agreements: Experiments in clouds and grids. In Cloud Computing (pp. 361-372). Springer, London.
3. Badger, L., Grance, T., Patt-Corner, R., & Voas, J. (2012). Cloud computing synopsis and recommendations. NIST special publication, 800, 146.
4. Bernstein, D. (2014). Containers and cloud: From lxc to docker to kubernetes. IEEE Cloud Computing, 1(3), 81-84.
5. de Alfonso, C., Calatrava, A. and Moltó, G., 2017. Container-based virtual elastic clusters. Journal of Systems and Software, 127, pp.1-11.
6. Balalaie, A., Heydarnoori, A. and Jamshidi, P., 2016. Microservices architecture enables devops: Migration to a cloud-native architecture. Ieee Software, 33(3), pp.42-52.
7. Boss, G., Malladi, P., Quan, D., Legregni, L., & Hall, H. (2007). Cloud computing. IBM white paper, 2007. 2009-9-18]. http://download. boulder. ibm. com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final _8Oct. pdf.
8. Bryant, R., Tumanov, A., Irzak, O., Scannell, A., Joshi, K., Hiltunen, M., ... & De Lara, E. (2011, April). Kaleidoscope: cloud micro-elasticity via VM state coloring. In Proceedings of the sixth conference on Computer systems (pp. 273-286). ACM.
9. Bui, T. (2015). Analysis of docker security. arXiv preprint arXiv:1501.02967.
10. Dejun, J., Pierre, G., & Chi, C. H. (2009, November). EC2 performance analysis for resource provisioning of service-oriented applications. In Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops (pp. 197-207). Springer, Berlin, Heidelberg.
11. Duala-Ekoko, E. and Robillard, M.P., 2007, May. Tracking code clones in evolving software. In 29th International Conference on Software Engineering (ICSE'07) (pp. 158-167). IEEE.
12. Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., & Concha, D. (2013). A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures. Future Generation Computer Systems, 29(1), 273-286.
13. Islam, S., Lee, K., Fekete, A., & Liu, A. (2012, April). How a consumer can measure elasticity for cloud platforms. In Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (pp. 85-96). ACM.

14. Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization. James Turnbull.
15. Salah, K. and Boutaba, R., 2012, November. Estimating service response time for elastic cloud applications. In 2012 IEEE 1st International Conference on Cloud Networking (CLOUDNET) (pp. 12-16). IEEE.
16. JoSEP, A.D., KAtz, R., KonWinSKi, A., Gunho, L.E.E., PAttERSon, D. and RABKin, A., 2010. A view of cloud computing. Communications of the ACM, 53(4).
17. Gao, J., Pattabhiraman, P., Bai, X. and Tsai, W.T., 2011, December. SaaS performance and scalability evaluation in clouds. In Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE) (pp. 61-71). IEEE.
18. Moore, L.R., Bean, K. and Ellahi, T., 2013. A coordinated reactive and predictive approach to cloud elasticity.
19. Pahl, C., Brogi, A., Soldani, J. and Jamshidi, P., 2017. Cloud container technologies: a state-of-the-art review. IEEE Transactions on Cloud Computing.
20. Garg, S.K., Versteeg, S. and Buyya, R., 2013. A framework for ranking of cloud computing services. Future Generation Computer Systems, 29(4), pp.1012-1023.
21. Hadar, E., Vax, N., Jerbi, A. and Kletskin, M., CA Inc, 2013. System, method, and software for enforcing access control policy rules on utility computing virtualization in cloud computing systems. U.S. Patent 8,490,150.
22. Hayes, B., 2008. Cloud computing. Communications of the ACM, 51(7), pp.9-11.
23. Herbst, N.R., Kounev, S. and Reussner, R., 2013. Elasticity in cloud computing: What it is, and what it is not. In Proceedings of the 10th International Conference on Autonomic Computing ({ICAC} 13) (pp. 23-27).
24. Hong, Y.J., Xue, J. and Thottethodi, M., 2011, June. Dynamic server provisioning to minimize cost in an IaaS cloud. In Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems (pp. 147-148). ACM.
25. Hong, Y.J., Xue, J. and Thottethodi, M., 2012, April. Selective commitment and selective margin: Techniques to minimize cost in an iaas cloud. In 2012 IEEE International Symposium on Performance Analysis of Systems & Software (pp. 99-109). IEEE.
26. Iosup, A., Yigitbasi, N. and Epema, D., 2011, May. On the performance variability of production cloud services. In 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 104-113). IEEE.

27. Iqbal, W., Dailey, M. N., Carrera, D., & Janecek, P. (2011). Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Future Generation Computer Systems, 27(6), 871-879.
28. Dua, R., Raja, A. R., & Kakadia, D. (2014, March). Virtualization vs containerization to support paas. In 2014 IEEE International Conference on Cloud Engineering (pp. 610-614). IEEE.
29. JoSEP, A. D., KAtz, R., KonWinSKi, A., Gunho, L. E. E., PAttERSon, D., & RABKin, A. (2010). A view of cloud computing. Communications of the ACM, 53(4).
30. Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl, V., & Tosun, C. (2012, August). Benchmarking in the cloud: What it should, can, and cannot be. In Technology Conference on Performance Evaluation and Benchmarking (pp. 173-188). Springer, Berlin, Heidelberg.
31. Li, Y., Zhang, J., Zhang, W., & Liu, Q. (2016, November). Cluster resource adjustment based on an improved artificial fish swarm algorithm in Mesos. In 2016 IEEE 13th International Conference on Signal Processing (ICSP) (pp. 1843-1847). IEEE.
32. Li, Z., O'brien, L., Zhang, H., & Cai, R. (2012, September). On a catalogue of metrics for evaluating commercial cloud services. In Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 164-173). IEEE Computer Society.
33. Liu, D., & Zhao, L. (2014, December). The research and implementation of cloud computing platform based on docker. In 2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP) (pp. 475-478). IEEE.
34. Li, W., Zhong, Y., Wang, X., & Cao, Y. (2013). Resource virtualization and service selection in cloud logistics. Journal of Network and Computer Applications, 36(6), 1696-1704.
35. Monsalve, J., Landwehr, A., & Taufer, M. (2015, September). Dynamic cpu resource allocation in containerized cloud environments. In 2015 IEEE International Conference on Cluster Computing (pp. 535-536). IEEE.
36. Möller, S., & Raake, A. (Eds.). (2014). Quality of experience: advanced concepts, applications and methods. Springer.
37. Bhattiprolu, S., Biederman, E. W., Hallyn, S., & Lezcano, D. (2008). Virtual servers and checkpoint/restart in mainstream Linux. ACM SIGOPS Operating Systems Review, 42(5), 104-113.
38. Youseff, L., Butrico, M., & Da Silva, D. (2008, November). Toward a unified ontology of cloud computing. In 2008 Grid Computing Environments Workshop (pp. 1-10). IEEE.
39. Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation

frameworks for cloud data centers. Journal of network and computer applications, 52, 11-25.

40. Watts, T., Benton, R., Glisson, W., & Shropshire, J. (2019, January). Insight from a Docker Container Introspection. In Proceedings of the 52nd Hawaii International Conference on System Sciences.