e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

ANALYSIS OF WEB ASSEMBLY TECHNOLOGY IN CLOUD AND BACKEND

Harsh Joshi[*1]

[*1]Department Of Computer Science & Engineering, Birla College Of Arts, Science & Commerce, Kalyan, India.

DOI : https://www.doi.org/10.56726/IRJMETS29640

ABSTRACT

Web assembly is a binary instruction format for stacked virtual machines. Web Assembly is a compilation target so that code can execute on browsers. With web assembly it is possible to achieve near native performance, speed for application on web and outside the web. It is found that it provides performance similar to languages like c,c++,rust [1]. The byte code it generates is fast to parse and compile for various platforms. Byte code is very light so less strain on the network for sharing. Edge computing is part of cloud computing made near data centres. The web assembly is first time used outside the browser in wasmRt was by [4]. In this paper the author is exploring various ways that Web assembly can be used outside the web browser and in docker and cloud, microservices and CMS. Using Web assembly outside the browser's environment creates various opportunities.

Keywords: Web Assembly, Docker, Containers, Virtual Machines, Edge Computing, Cloud Native, System Architecture, Etc.

I. INTRODUCTION

Nowadays every business is migrating towards cloud for scaling and management. Due to decrease in internet prices and increase in no devices, the internet services are now reached to everyone. So as people using the internet increased, as they want fast results, security and should be useful on every platform the cloud must become better. For making cloud native architecture better and services it provides faster we need something which goes beyond vm and containers and Kubernetes we started looking for some other solution. While in search of the next technology for cloud a lot of candidates appeared but web assembly attracted most of our attention. Web assembly promises small binaries, a secure sandbox environment in which the application has to ask permission explicitly, portable, and fast. Cloud enables user/developers to use various services to scale, maintain and monitor their application. There are mainly 3 giant companies in the cloud ecosystem which are google(gcp),Microsoft(azure), amazon(aws)(fig 1.1).Cloud provides a platform for various needed services as file storage, hosting etc, these all services are used by tech firms/organisation for increasing business. Cloud provides these all services using virtualization. Cloud mainly provides three services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In Business, speed and performance is very important.

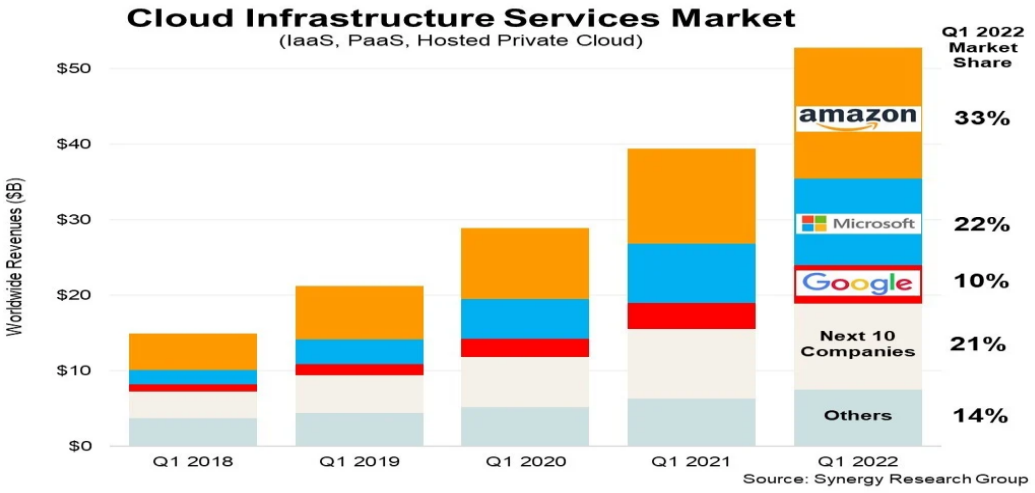

Figure 1.1: Synergy's Cloud Market share data forQ1 2022 [17]

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

In this paper, technologies like virtual machines(vm) , containers, Kubernetes(k8s) are discussed. Their pros and cons are analysed and how they complement each other as well as their comparison is performed. How entry of web assembly in cloud, backend, edge computing and in other cloud related fields changes the perspective of developers about cloud. Research and open-source projects are analysed to conclude the web assembly's impact and potential in the cloud.

This paper will help in further studies in the domain of web assembly and cloud as well as their merging for creating a better cloud in future.

II. BACKGROUND STUDIES

2.1. Virtualization

Virtualization is a process in which software is used to simulate hardware functionality and create a virtual computer system. In layman terms, Virtualization is a process in which a virtual computer system is created. Virtualization enables one to run and organise various operating systems and more than one vm on a system using hypervisor.

Virtual machines(vm) are virtual computer systems having their own operating system and applications inside it. Vm are emulated computer systems which run on a given operating system using hypervisor. Hypervisors are tools or programs to create, run and monitor virtual machines. There are mainly two types of hypervisors, one being BareMetal, running guest vm on system hardware directly and second being more traditional in which there is a host machine and guest vm runs as program on it, which can be Started and stopped as per need. Virtualization in the cloud helps for sharing infrastructure for systems.

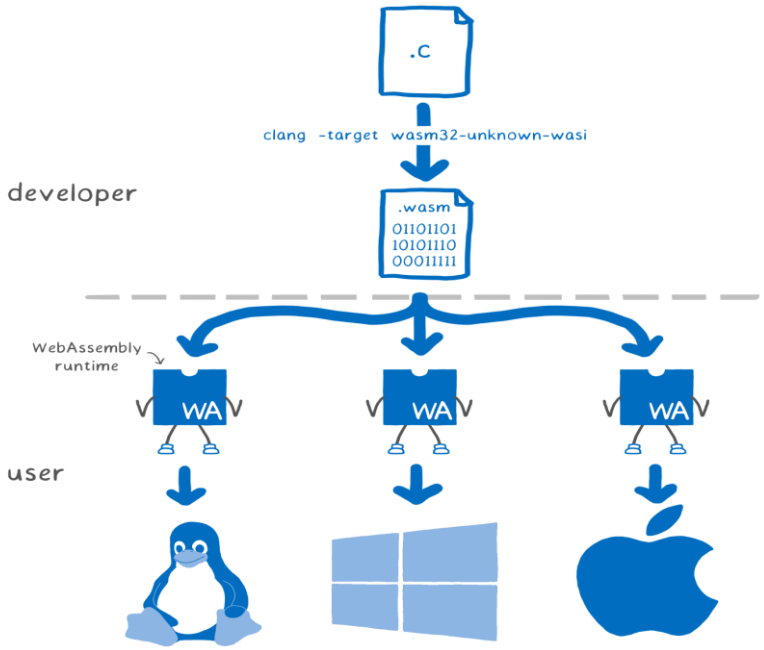

Figure 2.1: System for Running Web Assembly Outside the Web [16]

2.2. Containers

Containerization is a type of virtualization in which the system runs in its own, separate, isolated spaces called containers. Containers mostly use common operating systems. In the container there is isolation and encapsulation of all dependencies of application, needed binaries. With the use of containers there is no need for separate os as they all share the same kernel.

2.3. Web Assembly

Web assembly promises cross platform, cross architecture development, compact binary sizes, isolation sandbox [4] fast start up etc. But the web assembly features promised in web browser environments are also desirable in non-browser environments as well and it is possible [5]. Web assembly system interface(wasi) standardised wasm execution outside the browser [6]. The need for a better solution for cloud is speed and memory .The vm start-up time can be measured in minutes and image size is in gigabyte but the docker

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

container start time is in seconds and image size is in megabytes. Docker and containerisation have an edge over vm but there is always room for better technology. Web assembly can be used in the cloud but it should have functionalities for interacting with system memory and file systems.

2.3.1 Wasi (web assembly system interface)

For getting the near native speed the binaries should be as close as native instruction set architecture (isas) so a very low-level abstraction level should be created. As developers started getting the web assembly towards the server, they realised that they didn't just need an abstract isa they also needed an abstract operating system one that made it possible to run the same binary across a bunch of different operating systems while preserving the effectiveness of the web assembly sandbox. Wasi offers a set of standardised Api bindings web assembly to run outside. Web assembly imports the wasi api.

There are some concepts in wasi-

Hosts: hosts are web assembly runtime which runs web assembly module, previous they used to web engine like v8 but now as they don't have to deal with html, CSS, JavaScript lighter runtimes can be used such as wasmtime, lucet etc. Most web assembly runtimes can be used as a command line interface.

Guests: guests are web assembly modules which run on the hosts, hosts may provide additional functionality to guests by doing some functionality on its behalf.

This concludes, hosts can perform system level functionality like accessing the file system for guest modules using the wasi standardised Api. Wasi uses a capability-based security model so that guests have to implicitly ask for the permission from the host. As web assembly does not support multithreading yet the wasi in the network domain is not very successful. There are principles of design of wasi : compatibility-based security, interposition, capability and portability.

Some implementation of wasi are:

- wasmtime, Mozilla's Web Assembly runtime
- Lucet, Fastly's Web Assembly runtime
- A browser polyfill

But there are some limitations to wasi -

- Asynchronous I/O
- File watching
- File locking

Further research can be done in this field but due to limited resources and time these fields are not explored in this paper.

2.3.2 Serverless, Edge And Wasm

2.3.2.1 Serverless

Serverless is a service in which cloud provides resources on request, saving resources and being better economically as the system won't be idle when not in use. Serverless term is an unsuitably applied term, as developers don't create their own server but cloud creates for developer and manages. Serverless allows developers to develop software without concerns about the server also capacity planning, configuration, fault tolerance and scaling. Cloud manages the maintenance, scaling the server infrastructure. Serverless applications scale up or down as per the need. The application is in a container and the container is launched when needed on the cloud. Serverless application may fall into two services backend-as-a-service(baas) or Function-as-a-service (Faas)

Table below shows advantages and disadvantages of serverless

Tabel -1: Advantages and Disadvantages of Serverless

Advantages	disadvantages
Cost effective	performance
Elastic model (Easy to scale up and down)	Security (Less secure)
Increase in productivity (As no need to manage http calls)	Vendor lock in period

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

2.3.2.2 Edge / Edge Computing

Gartner defines edge computing as "a part of a distributed computing topology in which information processing is located close to the edge—where things and people produce or consume that information.". Edge computing brings data and computations near so that the real time device or software don't face any latency issues.

2.3.2.3 Edge Computing And Web Assembly

Murphy et al. [9] compared Wasmer (wasm+wasi runtime) with other serverless runtime. It showed that only naive c bested the Wasmer. In further tests aws was beaten by Wasmer, Wasmer was getting beat by only IBM and native c, so wasm+wasi can be used for edge computing.

Hall and Ramachandran [8] compare wasm and open-Wisk (open-source Apache serverless platform) using docker engine. They discovered wasm has better performance and also performs well with a cold start.

III. WASM IN CLOUD NATIVE SERVERLESS APPLICATION

Web assembly in the cloud and outside of the web browser is made possible with wasi. Features of web assembly made it desirable for implementation outside the cloud. Those features are small sizes of binaries, portable, fast and secure. Web Assembly is nowadays being used in the cloud to edge computing. Some use cases of web assembly in cloud native are envoy proxy, which is an open-source edge and service proxy. Envoy proxy has supported extensions as a form of custom filter to filter the traffic. Each cloud provider supports serverless computing and they have their own name for it, for example the amazon's aws calls its lambda [14] these are now started creating those in web assembly.

Until2022, cloud native computing foundation has received around 15-16 projects including WasmEdge Runtime, a cloud native Web Assembly Runtime; wasmCloud, a Web Assembly application framework; Krustlet, a pod in Kubernetes Tools for running Web Assembly programs in.

The server-side of Cloud native web assembly needs wasi to get os functionalities as well as Kubernetes support. The web assembly workload must alongside with other containers in Kubernetes cluster. For this we have two approaches kruslet ,which runs wasm project directly from Kubernetes pod and wasmedge's crunw , which is a direct replacement for Kubernetes runtime. Web Assembly not only controlled and managed by Kubernetes ,wasm can extend Kubernetes, Rafael Fernández López of SUSE introduced us to the kuberwarden project, which provides a strategy engine based on Web Assembly for Kubernetes [15].

With web assembly it is easier to integrate third party plugins in applications, centrally based cloud native applications can use Web Assembly runtime to safely and effectively execute any user-submitted code function. Shopify is embedding a wasm function to extend and customise it according to SaaS.

IV. VM VS CONTAINERS VS WASM FOR CLOUD

4.1 Vm:

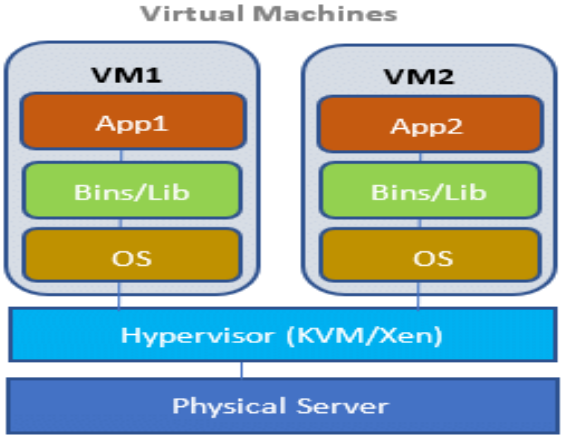

Fig 4.1.1: Virtual Machine underlying layers[15]

virtual machines are instances of a system running bare metal on hardware or running or host systems. Virtualization changed the cloud for the better. Virtualization can be defined as heavy weight, work horse for cloud. VMS run their own operating system and function separately which in return takes a lot of resources to manage which increases the cost of service from the cloud. Virtual machines are created using hypervisors,

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

hypervisors are code which helps in creating, managing and monitoring vm. Hypervisor's main role is providing an abstraction layer over hardware. Limits of vm: security-based attacks are easier to happen on vm as they share their resources and data, have possibilities of attacks due to vulnerability like buffer overflow, there is single point of failure in hypervisor type 1 as it has only one hypervisor and if it goes down the whole systems shuts off.

4.2 Containers:

container or containerization is a process of encapsulation and isolation of system binaries, software dependencies and libraries into one, to form a container. Coker containers are industry standard right now. Docker adds an additional layer of deployment engine over container engine. In core there are 4 main components of docker: deployment engine, docker image, docker container and docker client-server. Docker daemon/server and client can be running on same machines or local client can be connected [12] Docker images are snapshots of users file system, which creates an environment that gives the illusion of whole os to the software running inside. Containerization is very useful in cloud computing, Kubernetes or k8 is a platform used to execute docker containers in production clusters. K8s helps in creating cloud-native microservices applications. Kubernetes manages docker containers. Docker containers are faster, reliable and scalable.

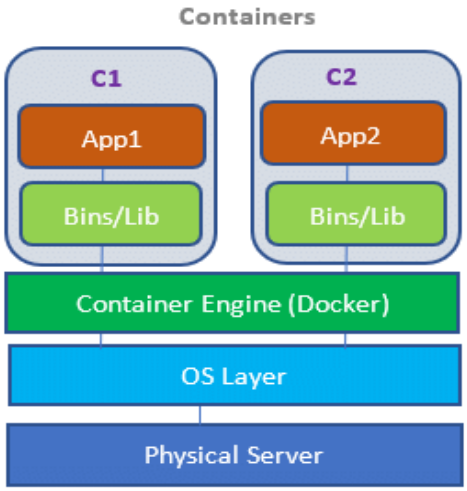

Figure 4.2.1: Container underlying layers[15]

4.3 Virtual Machines vs Containers:

In recent studies it has shown that containers are better than vm in cloud because of various reasons. Research done by Babak Bashari Rad, Harrison John Bhatti, Mohammad Ahmadi on docker shows docker is far better and faster than vm[10]. Also, research by Amit M Potdara , Narayan D Gb, Shivaraj Kengondc , Mohammed Moin Mulla [11] concludes that docker performance is far greater than vm ,stating qemu as one of the reasons that makes vm slower. But one paper states docker containers are not a replacement of vm but can be used along with each other for better results. Vm provides better Infrastructure as a service and Docker provides better SaaS (system as a service) [13]. So, we can assume that the docker container has a slight edge over vm.

4.4 docker container vs cloud native web assembly:

Solomon Hykes states in a tweet "if wasm+wasi existed in 2008, we wouldn't have needed to create docker....". That line describes the potential of web assembly in cloud and tech. Web assembly does not need to have a cold start like docker container so it is faster than docker as it does not need to start the os. The binaries of web assembly range between 1-10 mb but docker images can be of around 100mb size. But web assembly is not focused on replacing docker but working alongside it for better results. Matt Butcher, CEO of Fermyon shows example of docker and web assembly working together in The Finicky Whiskers.

https://github.com/fermyon/finicky-whiskers

We need web assembly in the cloud as it is fast. It takes mere Mili-seconds or microseconds to start the application in the cloud.We need web assembly for security as it prevents code from getting out of its sandbox and creating problems for other applications which are running in cloud

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

We need web assembly for cross platform uses such as code written once shall run everywhere, on every os and cross architecture use means it should run on intel, amd etc. Developers should have freedom to write software in their desired programming language.

V. EXPERIMENT, RESULTS AND DISCUSSION

Cncf(cloud native computing foundation) is part of the Linux Foundation ,they bring developers together to create open source software and host the largest conference. Cloud native empowers organisations to build and run scalable applications in the cloud. As web assembly in cloud native is a fairly new concept there are not a lot of projects made.

Rook : rook is cloud native storage orchestrated for kubernetes .It is graduated level cncf projects ,so it is in use by various consumers. Rook uses the facilities provided by the underlying cloud-native container management, scheduling and orchestration platform to perform its duties.

ContainerD : containerd is an industry-standard container runtime with an emphasis on simplicity, robustness and portability.

Atmos and Sat : Organisation named suborbital has created two products using web assembly : atmos and sat.

Atmo is an application framework for cloud native web assembly. The Atmo application framework consists of various functions from web assembly compiled from various languages. The core of Atmo is a job scheduler running WebAssembly modules, which allows running functions written in a variety of languages with near-native performance and massive improvements to security and ease of orchestration

Sat is a tiny webassembly server that runs one wasm file blazingly fast. Can run on smaller docker containers as well as bare metal.

5.1 Experiment overview.

I have created a simple API using typescript(type annotated javascript) rust and compiled rust code to web assembly. For typescript(ts) node js is used as runtime ,which is based on google's v8 engine. For rust i have used the Actix-web 3.1 web framework for handling http requests. Using cargo: rust's package manager, rust ocde can be compiled to wasm32-wasi.

Wasm binaries are run by using wasm engines like wasmtime, wasm-pack, Emscripten etc. wasmtime is used in this project. After conducting the experiment , conclusions can be derived from the results. My initial hypothesis is that web assembly will be better than the other two options . The experiment is mainly conducted to test speed , performance ,and security .

5.2 Experiment Setup

In this experiment Ryzen 5 4600U cpu,16 gb 3200MHz ram being used . The network's upload speed is 8MBps and download speed is 10mbps. Node version is v16.13.2, actix-web version 3.1. Code is written on vs code which is an ide.

Experiment is conducted with a simple API created in the backend in which a large payload is created from server side and delivered using get request and a large payload sent to backend using post request. The speed and performance is compared. The data need to be parsed before sending and receiving. The response time is noted. Now I have containerised this code and used those containers. created a lot of servers to emulate a busy and chaotic environment .The servers were single core so no multithreading was involved . Bucketed the results as good ,fine and bad response time and compared them at last. Good being between 15-20 ms. fine being 20-30ms and anything above that was considered bad.

5.3 Experiment Results and Analysis

After doing the experiment it is found out that rust is faster than node and ts ,But web assembly and rust had close competition but web assembly .Web assembly showed better result as expected.0

As rust and wasm had close results , there are some other factors contributing to wasm's better result in the experiment: the small size of compiled binaries, runnable on any architecture as well as platform. Due to the small size of binaries , doctor image formed was small as well, small docker container takes less memory and less resources making it a better choice for cloud as well.

Blue colour = less than 20 ms

Red colour = more than 20 ms and less than 30 ms

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

Yellow colour = more than 30 ms

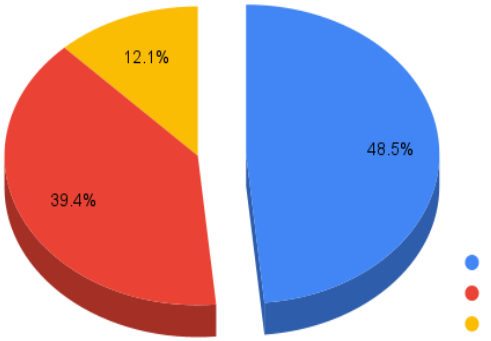

Figure-5.1: Typescript and Node js performance

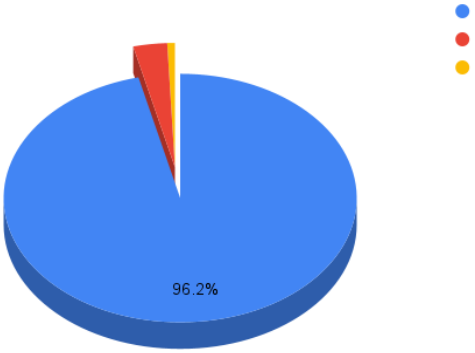

Figure-5.2: Rust Performance

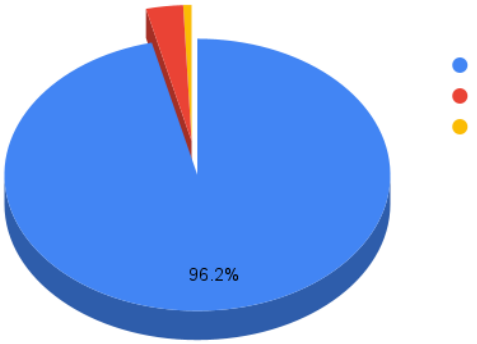

Figure-5.3: Rust and Web Assembly Performance

VI. FUTURE WORK

In future work the performance can be checked using multicores and multithreading. Using more resources from hardware. It can be used in AI. So after testing given programs and some other programs and software , technique ,framework the future of wasm in cloud native seems promising and a lot of potential to be found. These cloud solutions are deeply connected to docker and kubernetes .There is a lot of research to be done in this domain in the future scope. The web assembly and ai integration can be discussed in future scope

VII. CONCLUSION

In this paper the author discussed taking web assembly out of the browsers and applying it in various scenarios mainly cloud computing and edge computing and iot. As various research has shown that the web assembly provides better performance, small bite size, safety, safe sandbox, cross platform and cross architecture. Tools like wasi let the developer use os's file system and function for wasm. Difference and analysis of various cloud solutions like vm, docker and now wasm ,in this analysis it is found that docker containers are better than vm and wasm and docker containers can be used side by side using Kubernetes to get better performance in the cloud. As web assembly is very new in the cloud domain, various topics are not covered in this paper such as problems while using wasm in cloud native application.

VIII. REFERENCE

[1] Haas, Andreas & Rossberg, Andreas & Schuff, Derek & Titzer, Ben & Holman, Michael & Gohman, Dan & Wagner, Luke & Zakai, Alon & Bastien, Jf. (2017). Bringing the web up to speed with WebAssembly. 185-200. 10.1145/3062341.3062363.

[2] BEN L. TITZER , A fast in-place interpreter for WebAssembly ,arXiv:2205.01183v1 [cs.PL] 2 May 2022

[3] F. Eriksson and S. Grunditz, 'Containerizing WebAssembly : Considering WebAssembly Containers on IoT Devices as Edge Solution', Dissertation, 2021.

[4] Goltzsche, David, Manuel Nieke, Thomas Knauth and Rüdiger Kapitza. "AccTEE: A WebAssembly-based Two-way Sandbox for Trusted Resource Accounting." Proceedings of the 20th International Middleware Conference (2019): n. pag.

[5] B. Spies and M. Mock, "An Evaluation of WebAssembly in Non-Web Environments," 2021 XLVII Latin American Computing Conference (CLEI), 2021, pp. 1-10, doi: 10.1109/CLEI53233.2021.9640153.

[6] Jämes Ménétrey, Marcelo Pasin, Pascal Felber, and Valerio Schiavoni. 2022. WebAssembly as a Common Layer for the Cloud-edge Continuum. In Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge (FRAME '22). Association for Computing Machinery, New York, NY, USA, 3–8. https://doi.org/10.1145/3526059.3533618

[7] C. C. Spoiala, A. Calinciuc, C. O. Turcu and C. Filote, "Performance comparison of a WebRTC server on Docker versus virtual machine," 2016 International Conference on Development and Application Systems (DAS), 2016, pp. 295-298, doi: 10.1109/DAAS.2016.7492590.

[8] Adam Hall and Umakishore Ramachandran. 2019. An Execution Model for Serverless Functions at the Edge. In Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI '19). Association for Computing Machinery, New York, NY, USA, 225–236. DOI: http://dx.doi.org/10.1145/3302505.3310084

[9] Seán Murphy, Leonardas Persaud, William Martini, and Bill Bosshard. 2020. On the Use of Web Assembly in a Serverless Context. In Agile Processes in Software Engineering and Extreme Programming– Workshops, Maria Paasivaara and Philippe Kruchten (Eds.). Springer International Publishing, Cham, 141–145.

[10] Bashari Rad, Babak & Bhatti, Harrison & Ahmadi, Mohammad. (2017). An Introduction to Docker and Analysis of its Performance. IJCSNS International Journal of Computer Science and Network Security. 173. 8.

[11] Potdar, Amit & Narayan, DG & Kengond, Shivaraj & Mulla, Mohammed. (2020). Performance Evaluation of Docker Container and Virtual Machine. Procedia Computer Science. 171. 1419-1428. 10.1016/j.procs.2020.04.152.

[12] Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization.

[13] Yadav, Anuj & Garg, M. & Mehra, Ritika. (2019). Docker Containers Versus Virtual Machine-Based Virtualization: Proceedings of IEMIS 2018, Volume 3. 10.1007/978-9

[14] Abhishek, Manish. (2020). High Performance Computing using Containers in Cloud. International Journal of Advanced Trends in Computer Science and Engineering. 9. 5686. 10.30534/ijatcse/2020/220942020.

[15] https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/

[16] https://virtualizationreview.com/articles/2022/05/03/~/media/ECG/virtualizationreview/Images/2022/05/synergy_q1_2022.ashx