Measure The IoT Framework Using Docker With Fog Computing

K. Aruna, Dr. G. Pradeep

Abstract: Container virtualization technology is being developed to more effectively redefine the critical elements of IoT / Edge displays to enable the concept of device virtualization. A large number of IoT devices are connected to the Internet, due to this occurred huge data traffic and high delay. Although more resources can be deployed in the cloud to alleviate these load problems, this research proposes to use containers for virtualization with Lightweight Machine to Machine (LWM2M) communications and establish fog computing to extend the scalability of LWM2M. With the help of self-organizing functions, operators manage a wide variety of IoT service offerings, such as monitoring, sorting, retrieving, and managing profiles. According to the SOFT (Self-Organized Fog of Things) service approach, the blockage of services and devices involves network constraints. There are many different types of applications in the multichannel network. Fog is used not only to carry heavy loads, but provide less delay for critical applications. To create a container virtualization and connect the global IoT / M2M platform into a fog system architecture in which the intermediate nodes of M2M are highly scalable hierarchical container nodes are organized. Then, to make the IoT / M2M platform more scalable, use an algorithm for load balancing and dynamic measurement of service instances of intermediate nodes. Finally, measure the performance of our dynamic resizing mechanism using the "cAdvisor" tracking tool.

Index Terms: Docker container, fog computing, Fog of Things, Lightweight Machine to Machine, Performance Evaluation, Self-Organized IoT, Scalability improvement.

1. INTRODUCTION

Docker is a tool that helps you create, deploy and manage small, complete packages with all the information required to execute an application (code, libraries, runtime environment, system settings and dependencies). These packages are called containers. Each container is used with its own processor, memory, volume I / O and dedicated network resources, depending on the individual kernel and operating system. The way they share or allocate their resources varies considerably compared to docker virtual machines. These containers can be easily sent to other computers. However, container virtualization is more efficient because it does not have a guest operating system, and unlike virtual machines, it uses the host operating system, shared libraries, and related resources. Application-specific binaries and application-specific container libraries run on the host kernel, which speeds up processing and execution. Even launching a container takes only a fraction of a second. When all containers are shared, the operating system is hosted and contains only binaries and libraries associated with the application. It also ensures that the workflow is consistent for everyone involved, from development to deployment. You can easily measure the number of parameters and use the code easily. Container images can be stored centrally for each application. Images and repositories of locally created repositories are publicly available. Container scalability is a characteristic that a container application can handle with increased workloads. This can be achieved by restructuring the existing single-engine architecture to increase available resources or by providing additional containers within the cluster of distributed machines. With the demand of big data in large-scale IoT applications, fog is a new IT paradigm that adds a network edge between the node and the data center, enabling and moving big data. The system has positioned micro-data centers on the edge of a fog network, allowing it to compute, save, store and process data, reduce the amount of data being sent. Fog computing structures are fundamentally different from traditional cloud platforms: Fog computer platforms must rely on a large number of small points to provide computing resources close to any end user. Objects are connected to each other with networks, while the clouds are usually aligned with the most powerful data centers linked by high-speed dedicated networks. The rest of this papers are organized in this manner. Section II provides a content related works to docker and fog computing environment. Section III Introduces the background of container process. Section IV describes the computer technologies for flexibilities using simulation tools. Section V evaluates the performance of container with microservices. Finally, Section VI Completes the document and provides directions for future work.

- *K. Aruna is a Research scholar (part time) in Anna University, Chennai. And working as Assistant Professor, Department of Information Technology, A.V.C. College of Engineering, Mannampandal. Email: avcce.aruna@gmail.com.*
- *Dr. G. Pradeep, Professor, Department of Computer Applications, A.V.C.College of Engineering, Mannampandal. Email: pradeep.g8@gmail.com*.

2 RELATED WORKS

The Internet of Things (IoT) will connect more than 50 billion multimedia devices worldwide. As a high-density network that requires significant resources, new technologies are being developed that improve IoT performance. Fog is a new phenomenon that uses powerful and intimate nodes to help end users reduce latency, improving the consumption of resources and the quality of services. Similar software, Fog Computing is close to the IoT domain. The paradigm for delivering distributed cloud environments being able to work closely with physical devices and support important deadlines IoT services such as real-time data analysis. Extends current IoTPaaS solutions to network perimeter in a container-based approach. In particular, container virtualization offers new benefits multiplatform deployment allows general execution environments for cloud, edge / fog nodes and constrained devices. In fact, the same event that is virtualized through the container run efficiently on the edge and in the cloud. In addition, Containers can also work on devices characterized by a range of computer resources, such as the Raspberry Pi.

This feature ensures cross-functional mobility of resources devices are forced up to the cloud architecture. Another benefit introduced by the containers is the possibility of isolation, In the base system, all processes run on a virtualized container. This feature promotes deployment Multiple tenant sites because the same hardware can be shared among the different tenants. Container virtualization allows bundled software to run on multiple hardware configurations. Container technologies offer many advantages such as faster creation, instantiation, and initiation of virtualized instances, Compared to alternative virtualization solutions such as hypervisors. In addition, container-based systems benefit from greater usage / service allocation as virtual images are reduced [12]. These container features are best suited to the needs of IoT / edge screens. Fog system to increase the size of IoT / M2M platforms in the cloud. The fog is not just about leaving the cloud with a heavy load, but also provides a lower latency for critical applications. To expand the service, the first step is to integrate the global IoT / M2M platform into the fog. In fog system where OneM2M middle nodes are arranged in highly scalable hierarchical nodes based on containers. Industry 4.0 is the best use case to cover the fog. This is Industry 4.0, which has a significant number of sensors and actuators in the factory. With these sensors and connected actuators, manufacturers can easily create IoT (or) IIoT applications on their platforms [5]. Every data is collected from factory sensors is sent to the cloud for analytics processing. Nevertheless, this initial deployment revealed the real-time system response and scalability for such cloud-centric design should be improved. In [3], Nam Ky Giang, Rodger Lea et al. As the number of connected IoT devices increases, the amount of data generated by them devices are putting pressure on the sheer volume of data generated by these devices puts pressure on traditional centralized cloud computing infrastructure. System fog, resources are distributed closer to the edge network, which has become a computer model that supports many data-intensive or time-sensitive IoT applications [11]. Due to its widely distributed nature, fog IT infrastructures are often capable of handling large volumes of data generated by IoT applications on the edge network rather than transporting raw data streams to a remote cloud. In [5], Abdukodir Khakimov, Ammar Muthanna, edge computing will be used to develop and build the Internet of Things. Thus, computer edge data will severely impact the economies of the technology and telecommunications sector. The internet is currently moving towards a cloud-based architecture. Due to heavy data traffics, the exchange of high amount of data to the cloud has not only made the communication channel more difficult, but also delayed in transmission and a decline in the quality of data. Furthermore, with increase of the data traffic, support for mobility and geolocation is no less important. In [7], Koustabh Dolui, the presence of the edge computation layer enables data processing to be offloading and verified to the external layer. In addition, it enables collaborative computing between terminals and enables edge devices and data management policies. In [8], L. M. Vaquero et al. The fog system brings a lot of scenes together. Without involvement of third-party, pervasive computing and local servers can interact with each other and on a network to do processing and then store on it. These functions may be support for basic network tasks or new request and applications that operate in a sandbox environment. Users request the service portion of their devices to host these services and receive privileges to do so.

In [13], Cássio Prazeres et al. SOFT-IoT: The fog of self-organizing things. SOFT-IoT implements the concept of fog computing, part of data processing capabilities and service delivery functions are handled locally on "small servers", that is, near where the data is stored. SOFT-IoT enables algorithms to simplify the concept of computing and rely on critical functions executes on virtual machines, it is an alternative solution for centralized cloud computing approach. Like this, SOFT-IoT enables the operation at the cloud level; other data will be stored, processed and resolved complex operations. In other words, in SOFT-IoT, data processing and service delivery it happens locally to override current infrastructure Limitations and data processing, thus reducing the need for heavy computing resources on remote servers, means that the data is geographically far from where it is generated.

3 UNDERSTANDING THE PROCESS OF CONTAINER

Container is a simple virtual environment that’s own requirement data’s and system from all other containers. Containers use the same host system kernel, which provides better computational efficiency. Containers allow you to share binary files and access libraries with other containers, while hardware virtualization limits the range in a virtual machine environment. Because IoT devices are often low on resources, container technology enables them to use resources when sharing binary files and libraries that store space. Virtualization offers a number of advantages such as effective utilization of resource, increased system availability, reduced downtimes, better load balancing, isolation between running applications and cost savings. Reduced administration / maintenance. Virtualization can be further divided depending on the type of resources that need to be virtualized, such as hardware virtualization and operating system virtualization. In this type of virtualization act as a guest operating system and runs on the kernel. The link between the guest system and the main hardware is done through the compression layer of the hypervisor. Secure and independent service can be providing the virtualization, but it also causes significant overhead due to hardware emulation (communication overload between the guest operating system and host hardware). To reduce these overheads, another type of virtualization can be used; virtualization at the operating system level is also known as container virtualization [2]. Docker is an open source container processing platform that enables hosting and implementation of all types of software applications, platforms, middleware, databases, and compiled, developed and personalized software. In addition, the development, distribution, presentation and deployment of the software greatly facilitate the rapid maturity and stability of the docker platform. To run additional applications, create multiple containers on the same machine and run them simultaneously. It is used to avoid processor overhead. As the load increases, check the threshold value of the CPU and automatically create the new container to avoid overloading the CPU.

```
Docker-machine start

Set Master container (wrkr1, wrkr2, wrkr n)

Conn establish each service

for each in service:

        Build and run an image

        Allocate the job and execute at each container Repeat; (to service 'n') then
        Calculate container performance

#execute the command

# docker stats- - all   - - format "table{{.container}}  \t  {{.cpuperc}} \t {{.memusage}}"

Docker Swarm load balancing of requests to a simple service

if CPU<Threshold and RAM<Threshold:

        Run the container

elif CPU>Threshold and RAM>Threshold:
        Create a new container by calling RMI Service;

else:
        Stop the container.
```

Figure1.1 *Docker Container for Load Balancing algorithm*

The idea behind micro services is that some types of applications are easier to create and manage when broken down into smaller, collectible components. Each component is created individually and the application is a combination of its components.

4 COMPUTER TECHNOLOGIES FOR FLEXIBILITY

Fog is an extension of system that creates additional networking, extra space and computing paradigms on cloud servers and IoT components. However, these services are not fully networked group. Fog is primarily focused on simplifying applications that require a distributed computing approach low latency services, fog computing and even delay lessness support conscious services. The use of computation in the remaining time is obvious when the resources close to the user improve the overall service, unless the processing level is high. The fog that connects to the numerous nodes, it includes sensors and actuators on these nodes etc. Fog computing aims to move cloud operations (computing, networking and storage) to cloud, objects. Due to its location and proximity, the system fog provides a lower delay and faster response compared to the cloud. In a fog network, end users or devices can be communicated at the multi level hierarchical system. At the each level applying the filtering techniques and verify the specific data and determines whether to go to the next level for fog / cloud processing

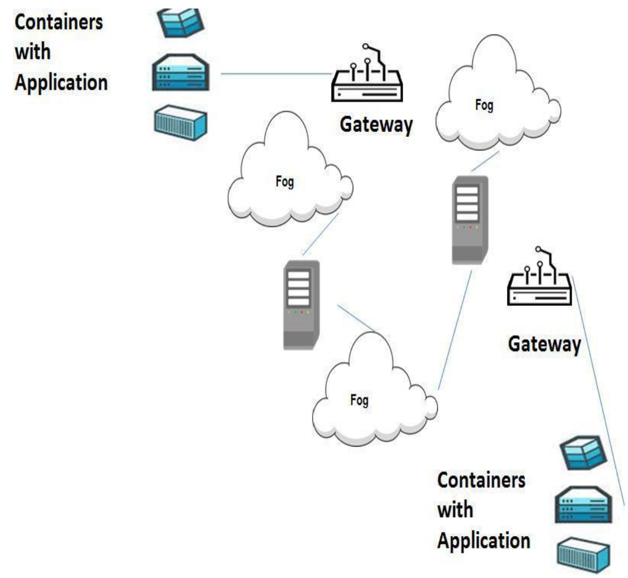

Figure1.2: *Fog Computing Container Model Architecture*

The fog paradigm joins the container for local-level FoT processing. Figure 1.2 shows detailed information about the container interaction between FoT. The FoT server must analyze the demand for the closest containers and the concentration of application results. Fog resistant devices cooperate with the gateway, requiring packages to be activated and compiled based on the Docker image for use.

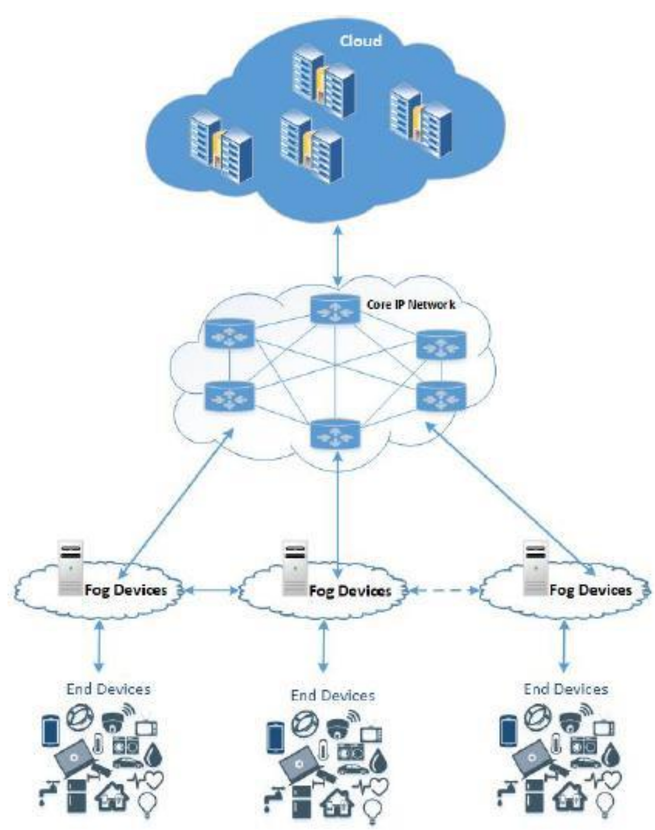

Figure 1.3: *Extension of Cloud system*

When compare the cloud and fog, the fog computation method achieves better scalability under various heavy traffic loads and the additional resizing does not create too much load with the server. The server can be automatically extending the scalability and processing. Compared with the standard approach, the efficiency of the system can be dynamically increased and reduced as needed. We use container-enabled virtualization to create a highly scalable M2M system based on fog. Containers provide a lightweight virtual environment compared to virtual machines, so are more suitable for quick response on a fog network. With this system fog provides an advance in cloud computing. This will reduce data traffic; avoid data loss and quick response. Access data easily, anytime and anywhere.

Various fog simulations tools and their main characteristics:

Name of the simulation tool	Proposed by	Description
CloudSim	Calheiros et al. [9]	A broad simulation toolkit that enables simulation and modeling of application provisioning in the cloud computing environments. The CloudSim toolkit supports system modeling of cloud system components such as virtual machines (VMs), data centers and resource allocation and provisioning policies as well as support system behavior modeling.
iFogSim	Gupta et al. [10]	Modelled IoT and Fog environments and measure the impact of resource management techniques in terms of network congestion, latency, cost and energy consumption. Developed on top of CloudSim.

Figure 1.4: *Fog Simulation tools*

In this module, we have used a simulator is called iFogSim for modeling IoT and fog environments and quantifying the impact of resource handling techniques on delay, network congestion, power usage and expenses. Design an IoT / Fog environment in iFogSim to analyze applications performance and effective resource utilization strategies. iFogSim is a high-performance open source toolbox for Fog, edge computing and IoT. This allows modeling and simulation of the fog system environment to estimate delay and resource management time. It incorporates more customizable resource management techniques based on the research area. It works with CloudSim, which is widely used in cloud environment simulation and resource management.

Simulation using iFogSim

Here are the classes of iFogSim needed to simulate the fog network:

- Fog device
- Sensor
- Actuator
- Power Monitoring
- Monitoring components
- Resource management service

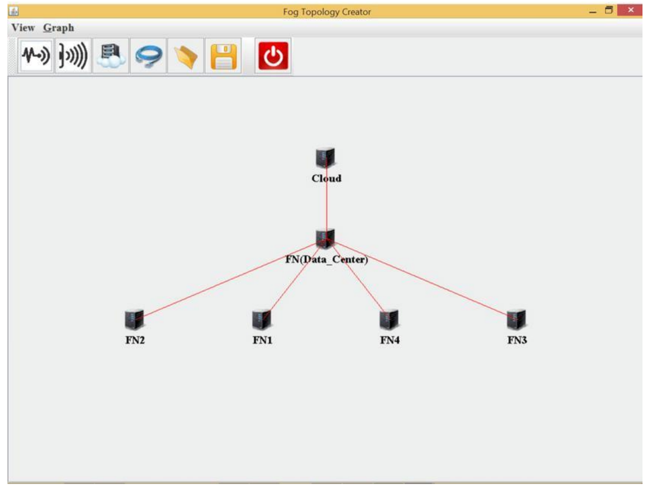

Figure 1.5: *Fog Topology Creator*

5 PERFORMANCE EVALUATION

The main parameters of container performance analysis are CPU, memory, volume I / O and availability. In a standard menu container, a large number of processes or applications can run simultaneously. Alternatively, to design an application with multiple containers, it can communicate via a connection system. This guarantees an application that there is no conflict between devices at runtime.

Performance between single containers with multiple containers can be distinguished by graphical representation.

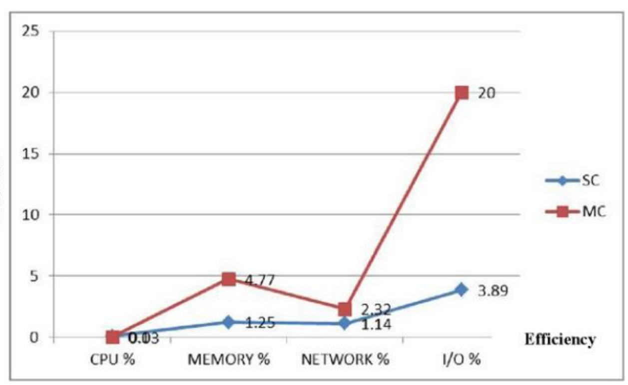

Figure 1.6: *Performance comparisons of single container and multiple containers*

In individual containers, performance for the same application is clarified. Container performance comparisons are illustrated in Figure 1.6, when the container is blocked from a single container, which differs from the host device, and only originates through the cluster. Consultant is an open source container resource utility collector. Native support for docker containers and support for other types of containers. cAdvisor works through the node.

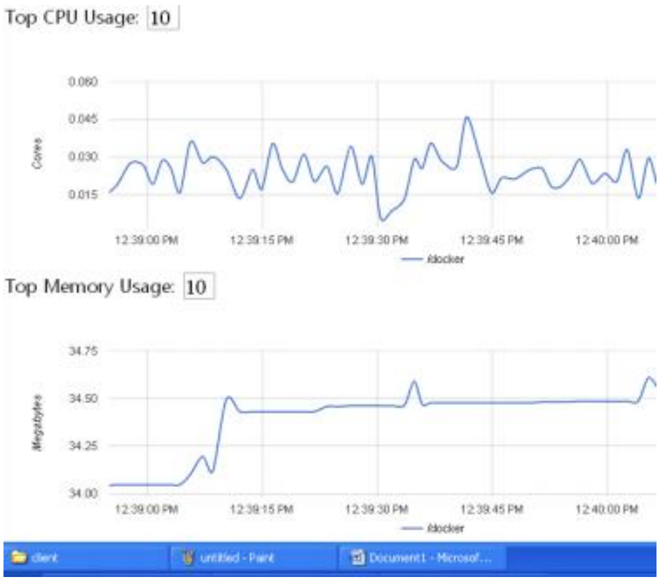

Figure 1.*7: Performance Evaluation of Container*

It automatically detects all containers in a given node and collects CPU usage, memory, file system and network usage statistics. Container, standard menu includes machine container usage and its dependencies compared to virtual machines. It runs an application process in the user space of the host operating system; share the kernel with other containers. In addition, it expands the self-regulated IoT network and scalability and improves the performance.

6. Conclusion and Future work

The container must maintain the need for multitasking in all operations, so that single-use, single-container and stand-alone devices are not sufficient for the heavy load of one application. Container approaches help container demand for many future prospects as container enhanced multitasking. The proposed method is designed by the container load balancing algorithm and allows the nodes to be evaluated at different levels of fog to get better performance of the network by allocating node resources to different tasks. Nodes can choose best relatives neighboring nodes and shared resources or perform a task. Finding and deploying the best neighbors, various kinds of techniques can be used in different tasks; using multiple techniques in the fog can reduce overhead costs. Improve the performance of the dynamic resource system. Container virtualization is the best solution for deployment container operating system virtualization supports microservices and all dependencies in one image. A container can manage the physical hardware resources required for an application with its operating system kernel applications. Future research tasks in fog-based networks include security, privacy, and integrity.

ACKNOWLEDGMENT

We would like to thank all the reference authors for providing access to the information and images on their IoT environment.

REFERENCES

[1] Robert Morabito, “A Performance Evaluation of container technologies on Internet of Things devices”, IEEE conference on Communications Workshops (INFOCOM WKSHPS) pp. 999-1000, 2016.
[2] Roberto Morabito, “Virtualization on Internet of Things Edge Device with Container Technologies: A Performance Evaluation”, IEEE Access, 2017, Volume 5, pp 8835-8850.
[3] Nam Ky Giang, Rodger Lea, Michael Blackstock, and Victor C.M. Leung, “Fog at the Edge: Experiences Building an Edge Computing Platform”, 2018 IEEE International Conference on Edge Computing.
[4] Chih-Lung Tseng and Fuchun Joseph Lin, “Extending Scalability of IoT/M2M Platforms with Fog Computing”, 978-1-4673-9944-9/18/$31.00 ©2018 IEEE.
[5] Abdukodir Khakimov, Ammar Muthanna, “Study of Fog Computing Structure”, 978-1-5090-4865-6/17/$31.00 ©2018 IEEE.
[6] Ashkan Yousefpour, Genya Ishigaki, and Jason P. Jue, “Fog Computing: Towards Minimizing Delay in the Internet of Things”, 2017 IEEE 1st International Conference on Edge Computing.
[7] Koustabh Dolui, “Towards Multi-container Deployment on IoT Gateways”, 978-1-5386-4727-1/18/$31.00 ©2018 IEEE.
[8] L. M. Vaquero and L. Rodero-Merino, “Finding your way in the fog: Towards a comprehensive definition of fog computing,” ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 27–32, 2014.
[9] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya,“Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.
[10] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, “ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments,” Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, 2017.
[11] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1–1, 2017.
[12] R. Morabito, J. Kjällman, and M. Komu, ``Hypervisors vs. lightweight virtualization: A performance comparison,'' in Proc. IEEE Int. Conf. Cloud Eng. (IC2E), Mar. 2015, pp. 386_393.
[13] Cassio Prazeres and Martin Serrano, “SOFT-IoT: Self-Organizing FOG of Things”, cas sio.prazeres@insight-centre.org, Brazil, 2016.
[14] Ruchika, “Evaluation of Docker for IoT Application”, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169, Volume: 4 Issue: 6.
[15] Pankaj Mendki, “Docker container-based analytics at IoT edge”, 978-1-5090-6785-5/18/$31.00 © 2018 by IEEE.
[16] Ranesh Kumar et al, “Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions”, arXiv:submit/2317175, July 2018.
[17] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, “An updated performance comparison of virtual machines and linux containers,” in 2015 IEEE International Symposium

on Performance Analysis of Systems and Software (ISPASS). IEEE, mar 2015.

[18] J. Shetty, S. Upadhaya, H. Rajarajeshwari, G. Shobha, and Chandra, “An empirical performance evaluation of docker container, stack virtual machine and bare metal server,”.

[19] A. Lingayat, R. R. Badre, and A. K. Gupta, “Integration of linux containers in openstack: An introspection,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 12, no. 3, Dec. 2018.

[20] Ab Rashid Dar “Fog Computing: An Efficient Platform for the Cloud-resource Management”, JETIR, Vol 6, Issue 2, Feb 2019.

[21] S.Balamurugan “Fog Computing: Synergizing Cloud, Big Data and IoT- Strengths, Weaknesses, Opportunities and Threats (SWOT) Analysis”, IRJET, Vol 3, Issue 10, 2016.