

# Lightweight Computation to Robust Cloud Infrastructure for Future Technologies

Sonia Shahzadi<sup>1</sup>, Muddesar Iqbal<sup>2</sup>, Xinheng Wang<sup>3</sup>, George Ubakanma<sup>4</sup>, Tasos Dagiuklas<sup>5</sup>, and Andrei Tchernykh<sup>6</sup>

<sup>1</sup> Swan Mesh Networks Ltd, Research and Development, London, UK

<sup>2</sup> University of Essex, School of Computer Science and Electronic Engineering, UK

<sup>3</sup> Department of Electrical and Electronic Engineering, Xian Jiaotong University, Suzhou, China

<sup>4,5</sup> School of Engineering, London South Bank University, UK

<sup>6</sup> CICESE Research Center, Ensenada, Baja California, Mexico

**Abstract.** Hardware and software lightweight solutions became the mainstream for current and future emerging technologies. Container-based virtualization provides more efficient and faster solutions than traditional virtual machines, offering good scalability, flexibility, and multi-tenancy. They are capable of serving in a heterogeneous and dynamic environment across multiple domains, including IoT, cloud, fog, and multi-access edge computing. In this paper, we propose a lightweight solution for LCC (Live Container Cloud) that permits the user to access live/remote cloud resources faster. LCC can be embedded as a fog/edge node to permit the users to allocate and deallocate cloud resources. The performance of such a containerization technology is presented.

**Keywords:** Cloud computing · Internet of Things · Fog Computing · Docker · OpenStack.

## 1 Introduction

Cloud Computing is shifting from large centralized data centers to the distributed multi-cloud environment to provide more efficient services. It can be integrated into Internet-of-Things (IoT) devices, edge, and fog computing. To provide computational power, storage capability, reliable connectivity, and other resources for billions of devices, we need lightweight solutions to be run on resource-constrained devices. They should be feasible for smart infrastructures and provide elasticity and exibility. We have proposed a lightweight virtualization solution for cloud services that is more efficient than traditional VMs. Lightweight virtualization based on the container has gained popularity for cloud deployments due to containers are more light and portable compared to VMs [5]. A VM is an emulation of a computer system with the independent OS, while containers are based on the host operating systems and share the same kernel with other containers running in the same machine. That is why containers require fewer resources and smaller in size. It makes them more suitable for resource-constrained devices [6]. Docker is one of the most popular solutions for container

deployment and management, providing exible, extensible, and portable computation [2] [2]. Linux Container (LXC) LXC is a well-known set of tools, templates, library, and language bindings.

In this paper, we present a lightweight, portable infrastructure solution based on LXC to enable on-demand resource provisioning. We provide a performance evaluation of the hypervisors and answer the following questions

1. Which cloud infrastructure is suitable for the public and private sectors?
2. Should we continue to use traditional VM infrastructure on the cloud or move to lightweight technologies for fast robustness?
3. Which type of virtual servers is suitable for future technologies?
4. How do these tools help to provide service migration on future generation technologies?

## 2 Literature Review

In the last era of computing, virtualization technology plays a vital role, especially in cloud computing. After the evolution of virtualization, the demand for these virtualized resources is increased. Two popular types of virtualization technologies are hypervisor-based virtualization and container-based virtualization [3]. VM can snapshot the whole working environment, including software, applications, dependencies, and more [4]. These snapshot images can be hosted on other servers. This approach facilitates reproducibility [4]. Virtualization is the key part of cloud services as cloud computing provides virtual resources on user demand [1].

This virtual environment can be deployed on physical machines and is completely transparent to the clients.

### 2.1 Hypervisor based virtualization in Cloud Computing

A hypervisor or Virtual Machine Monitor (VMM) is executed on a host or a physical system by splitting and allocating host resources into a guest operating system or VM. There are two types of hypervisor-based architectures. In Type 1, the hypervisor runs directly on the top of the host hardware, while in Type 2, the hypervisor runs on the top of the host operating system. Several hypervisors are used according to deployment and optimization such as KVM, QEMU, Microsoft Hyper-V Server, etc. They provide the following features:

- *Encapsulation*: A virtual hard disk is located in VM to store les. This disk can be easily moved, backed-up, copied, replicated or migrated from one host to another one.
- *Isolation*: User can run multiple virtual machines on the same host without interfering. Each VM has its operating system and software. Failure of one VM does not affect other VM or host operating system.

- *Transparency*: VM and guest operating system are unaware that they are running in a virtual environment and programs run within the virtual environment in the same way as they run on physical machines.
- *Manageability*: Hypervisor is responsible for allocating resources to VMs and execution of VMs through a management interface.

## 2.2 Container-based virtualization in Cloud Computing

In container-based virtualization, all containers share the same operating system, which reduces the runtime and storage overhead. A container image provides a standalone package for container instantiation and execution. It encapsulates everything, including system configuration and applications [8]. Multiple containers can be executed on the same physical server or different servers to provide fault tolerance. Although containers share the same operating system, they are isolated from each other through process IDs, inter-process communication, network interfaces, and directory trees. Control Groups (cgroups) are used for resource management, while namespaces are used for processes isolation. Resource optimization is achieved by providing low overhead with transparency.

## 2.3 Comparison of VM and Container

OpenStack is one of the open source solutions for cloud implementation that is widely used for private and public sectors. Hypervisors are responsible for providing services for clients. OpenStack mostly used KVM hypervisor. However, Google and IBM are successfully using containerization technology [1]. The main features of VMs and containers are given in Table 1, and in Figure reffig1. They show the importance of Docker container adoption.

**Table 1.** Comparison of VM and Container

S.No	Features	VM	Container
1.	Isolation	H	L
2.	Resources wastage	H	L
3.	Transparency	H	H
4.	Runtime overhead	H	L
5.	Storage overhead	H	L
6.	Fault Tolerance	L	H

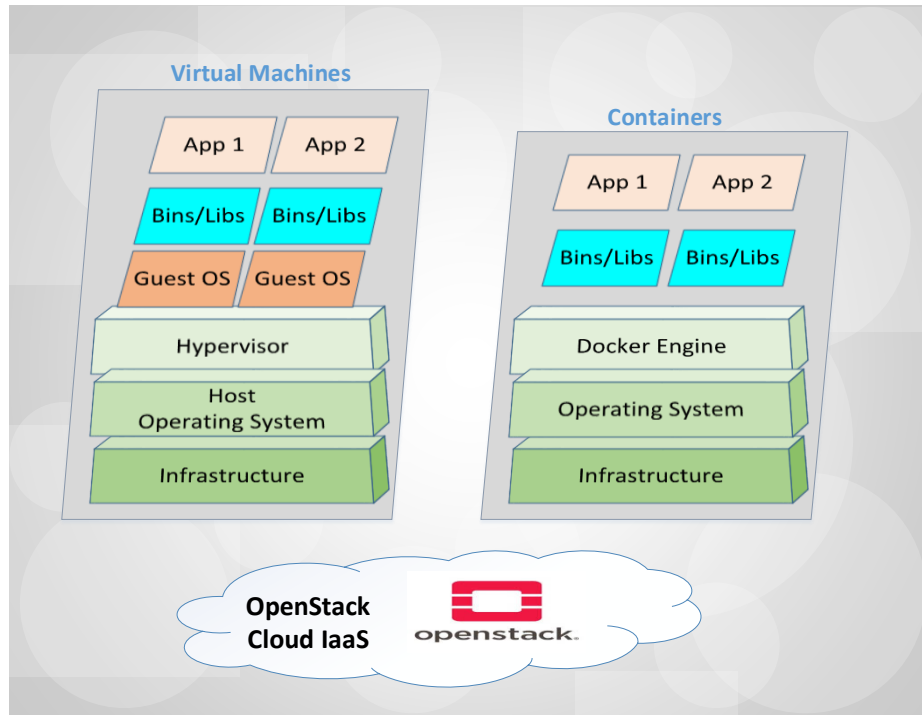


Fig. 1. Virtual Machines vs Containers

### 3 Use Cases

Millions of devices with sensing and actuators capabilities are connecting with the Internet of Things (IoT) in everyday life, and mostly IoT objects send data to cloud servers for computation purpose and then back to the devices. However, this method is not sustainable due to the rapid growth of IoT objects as a massive quantities of devices creates congestion, security, and latency challenges. Fog Computing introduces a new paradigm to overcome these challenges where edge nodes/devices are deployed at the edge of the network for local capabilities. These edge nodes are resource constraints and varying according to processing and storage. Cloud computing provides a lightweight solution with a container for distributed applications deployment where computation infrastructure can be robust for smart services. Figure 2 demonstrates the proficiency of LXC (Linux Container) for two cases.

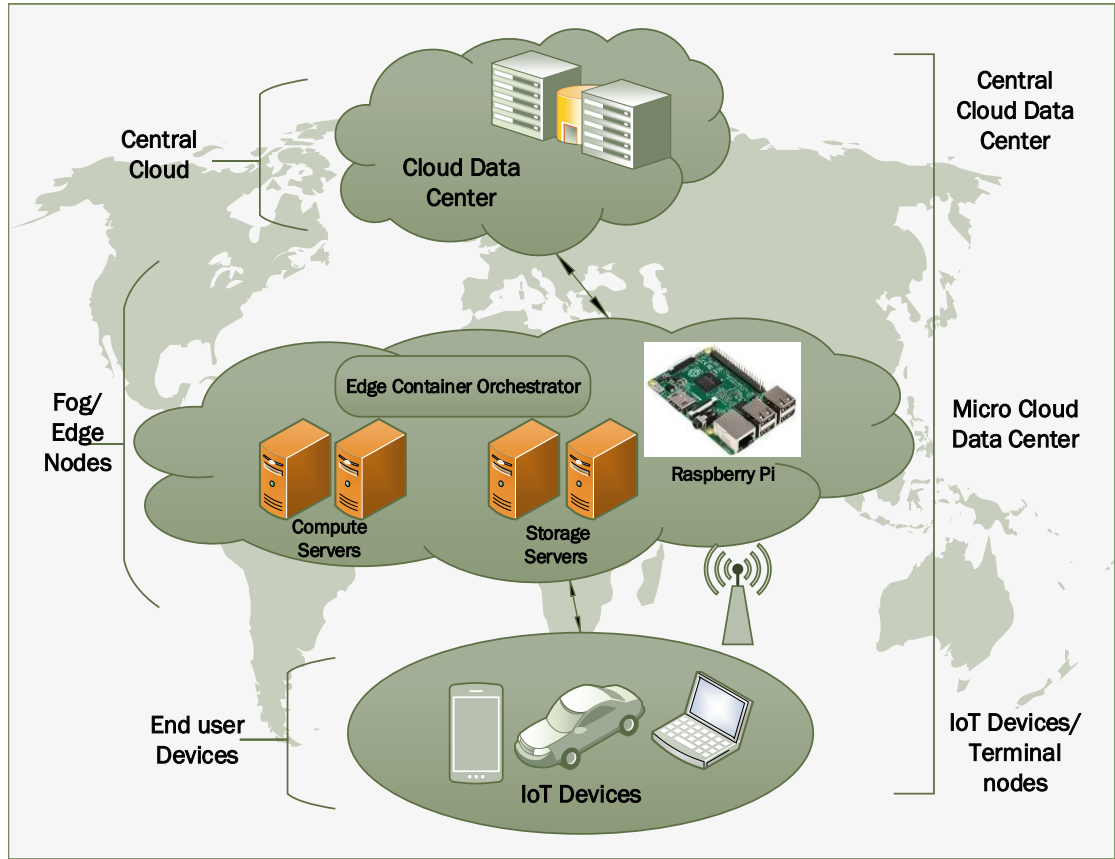
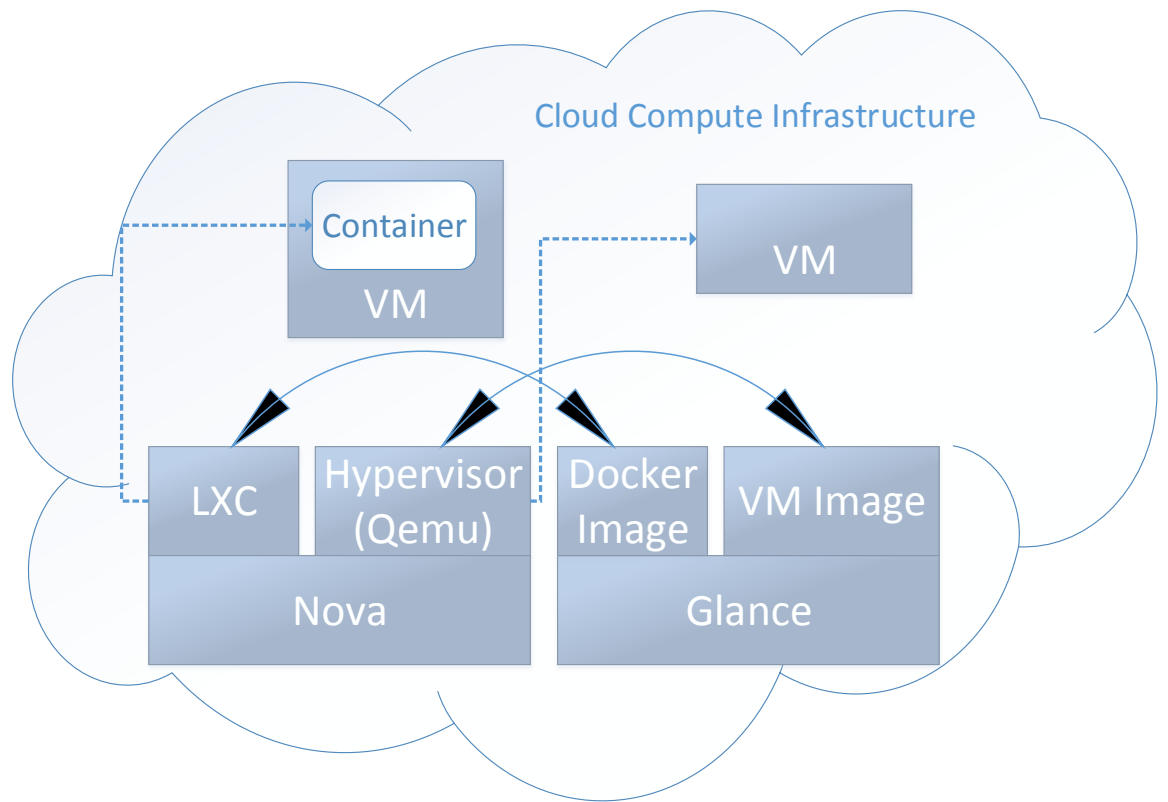


Fig. 2. Lightweight Architecture for Future Technologies

## 4 Implementation

To implement lightweight technology inside the cloud infrastructure, we have the following steps:

- **Configure Cloud IaaS to enable Docker:** To enable Docker as a hypervisor in OpenStack cloud, we have configured this with compute node and performed several tests to prove its functionality for a production-ready environment.
- **Initiated via Docker Hypervisor:** In this scenario, we have compared multiple hypervisors in cloud computing scenarios where we can compare the performance of cloud servers via Qemu and LXC, as shown in Figure 3.
- **Live cloud resources deployment from container:** In this scenario, we have compared the performance of live cloud resources that use containers and VMs (Figure 4).



**Fig. 3.** Docker as a Hypervisor

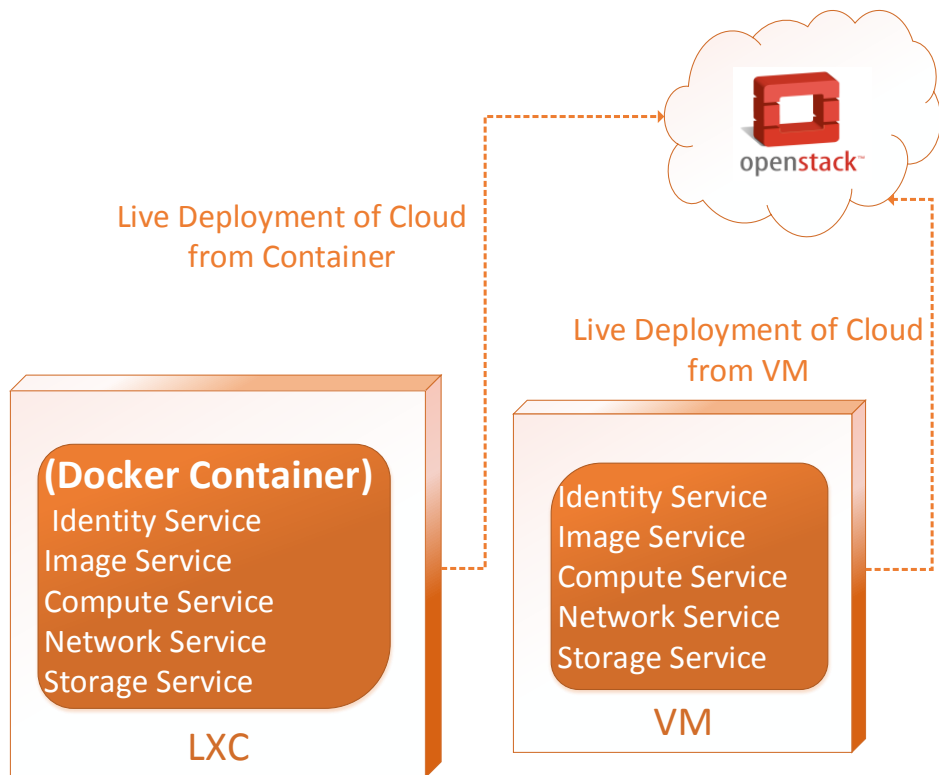


Fig. 4. Cloud Resources access from Container

## 5 Performance Evaluation

To evaluate the performance of VM and container, we setup a cloud testbed using OpenStack and Docker container integrated with OpenStack. Ubuntu 14.04 is used as a host OS with 3-tier cloud architecture. We have measure VM hypervisor and LXC performance in following two scenarios.

### 5.1 Instance/ Server Performance Measurement

We have performed these tests under the following environments.

Cloud Platform: OpenStack

Host OS: Ubuntu Server 14.04

VM hypervisor: QEMU

LXC: Docker container

Image: cirros-0.3.3-x86\_64 (12.6 MB), tutum/wordpress (485.5 MB)

Image format: qcow2, raw

Docker images are lightweight, but we have chosen 485.5 MB size image for Docker and 12.6 MB size for Qemu. We have started and deleted servers with both VM and Docker images. Booting and deleting times are shown in Figure 5. We see that Qemu VM is about 3.5 times slower.

### 5.2 Cloud Performance Measurement

To compare cloud performance of VM vs. Docker, rst, we have deployed cloud services from VM and measure parameters. Then, we have deployed live cloud services from the container. Figure 6 shows that the containerized solution is more than twice as fast as VM. Parameters of both approaches are presented in Table 2.



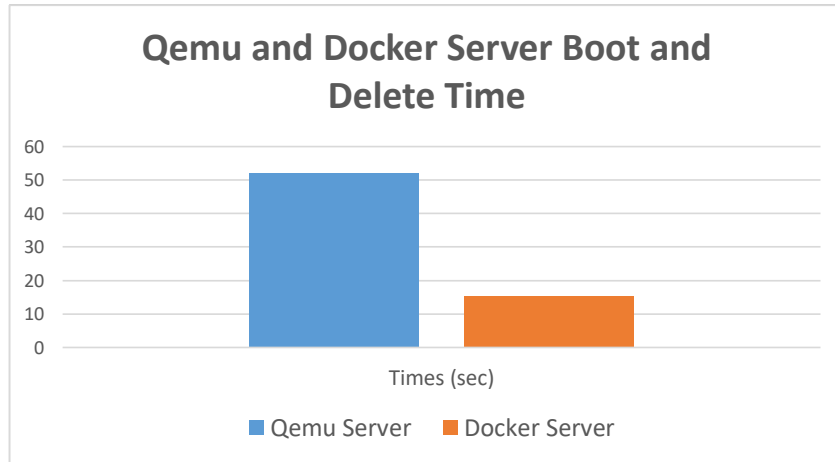
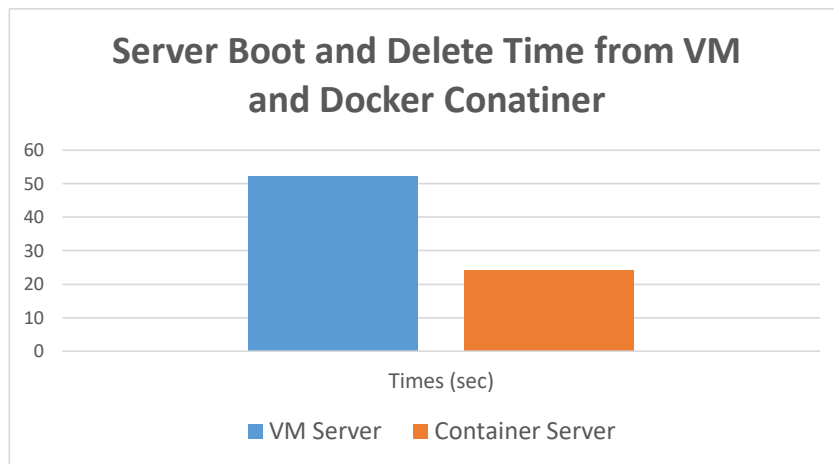


Fig. 5. Docker vs VM server boot and delete time

Table 2. Comparison of results

Performance Measurement	Cloud Platform	Hypervisor	Image Name	Image Size	Format
Hypervisor Performance Measurement	OpenStack	Qemu	Cirros	12.6 MB	Qcow
		Docker	Docker Cirros	485.5 MB	Raw
Cloud Performance Measurement	OpenStack	VM	Cirros	12.6 MB	Qcow
		Container	Cirros	12.6 MB	Qcow



**Fig. 6.** Cloud Performance Measurement from Container

## 6 Conclusion

The number of personal devices such as laptops, smart-phones, tablets, etc. are growing rapidly. They are constrained in terms of computational power, storage, connectivity, power supply, and other resources. It creates a heavy load on IT infrastructure and needs efficient lightweight solutions to deliver on-demand resources provisioning. To support robust and portable computation processes and applications, Linux container plays a vital role. Containerization technology enables to pack more applications into a single physical server compares to VM. When we consider migration, containers take less time compared to VMs, as they require fewer system resources. We have developed the testbed to measure the performance of the proposed solution. We compare VM and container booting and deleting in several scenarios using OpenStack and LXC. Obtained results clearly show that container solution is better as compared to typical VM hypervisors in the cloud computing environment.

## References

1. Bernstein, D.: Containers and cloud: From lxc to docker to kubernetes. *IEEE Cloud Computing* **1**(3), 81–84 (2014)
2. Boettiger, C.: An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review* **49**(1), 71–79 (2015)
3. Bui, T.: Analysis of docker security. arXiv preprint arXiv:1501.02967 (2015)
4. Howe, B.: Virtual appliances, cloud computing, and reproducible research. *Computing in Science & Engineering* **14**(4), 36–41 (2012)
5. Pahl, C.: Containerization and the paas cloud. *IEEE Cloud Computing* **2**(3), 24–31 (2015)
6. Soltesz, S., Pötzl, H., Fiuczynski, M.E., Bavier, A., Peterson, L.: Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In: *ACM SIGOPS Operating Systems Review*. vol. 41, pp. 275–287. ACM (2007)