Hybrid Run Time Offloading and Resource Allocation in Mobile Assisted Cloudlet Based Cloud Network

Fida Hussain Khoso[1], Abdullah Lakhan[2], Aijaz Ahmed Arain[3], M. Ali Soomro[3], Shah Zaman Nizamani[3] , Zahoor Hussain[4]

***Abstract*— This paper presents a novel Mobile Assisted Cloudlet Based Cloud system by using container technology. The main objective of the plan is to offer a hybrid runtime environment to run OCR, Augmented Reality, and 3D gaming applications on different computing models. The work devises a novel container technology-based hybrid runtime aware framework which divides the applications as mentioned earlier into different computing environment to minimize the response of applications. To solve the problem, the work suggests a latency aware Task Assignment (LATA) algorithm framework, which runs the applications as mentioned earlier via different steps. The performance evaluation shows that the proposed system and algorithm outperform existing monolithic virtual machine systems in term of the response time of applications.**

Keywords—: Resource Allocation, Mobile Cloud, Cloudlet, Container, LATA.

INTRODUCTION

These days, many web applications are converting into mobile cloudlet based applications. However, latest portable devices have resource-constraint issues such as memory, storage and processing and cannot run these applications stand-alone locally [1]. Offloading is a technique which transfers heavyweight workload of applications to the centralized cloud for execution [2]. However, centralized cloud locates multiples hops away from devices and incurs with long end to end latency during performing [3]. The recently emerging cloudlet is a new paradigm which brings cloud services at the edge of mobile network [4]. The mixture of mobile computing, cloudlet and cloud-based architecture is an effective way to run different classes type applications such as delay sensitive, delay-tolerant at other models [5]. Many existing studies suggested different mobile cloudlet based cloud architectures to execute different classes of applications [6-10]. The design runtime based on virtual machine cloning which can execute the entire workload of applications. The runtime here means the environment on each computing model which can understand tasks binaries and execute them under their extensions. For instance, Android X86 based architectures support all android applications at a different platform. However, existing virtual machine-based hybrid runtime faces many challenges. For example, virtual machines have long setup time, overhead among virtual machines, and pre-allocation of resources for the prior offloaded workload [11, 12]. This paper proposes a novel container-based hybrid runtime framework. The framework leverages mobile computing, cloudlet, and cloud combine to execute all class of applications the goal of the framework to divide the workload of applications into different platform according to their capacities. The framework divided the application into mobile tasks, cloudlets tasks and cloud tasks at the design of applications. Each workload of application has these attributes (data size, execution time, and deadline). The different vendors offer container-based services to minimize the response time of applications. The work has the subsequent research benefactions as explain below.

Improve the hybrid offloading-performance, the novel hybrid run time framework has been proposed by our study.
To understand the problem behaviour, a novel mathematical model describes the problem behaviour with their given constraints.

Transmission delay has thumping influence over offloaded components, to improve the offloading components migration performance, a mid-sort Q-learning algorithm has proposed.
Latency sensitive and highly responsive application (i.e., 3D Game) has a fixed latency bound; to allocate requested workload within a given latency bound an iteratively based novel LATA algorithm has proposed. To execution time of application must be less the given deadline.

The remainder of the paper organizes as follows: Section 2 discussed related work. Section 3 defines the problem formulation. Section 4 shows the proposed framework and algorithms in detail. Section 5 shows performance evaluation of the proposed schemes and result in the discussion of all workloads. Section 6 tells about the conclusion of the paper with proposed methods and future work of the study.

RELATED WORK

These days, many web applications are converting into mobile and smart applications. Due to hybrid runtime, it encourages to run one mobile application into different computing nodes due to its device resource constraint issues the runtime environment facilities for running application other computing nodes with their extensions. Many efforts have been made in the literature solve delay optimal problem via different

[1]Dawood University of Engineering & Technology
[2]Southeast University Nanjing, China
[3]Quaid-e-Awam University of Engineering, Science and Technology
[4]Indus university Karachi.
Email: fidahussain.khoso@duet.edu.pk

runtime architectures.

These studies [1- 5] proposed a new runtime architecture by exploiting mixture mobile and cloud computing to run delay-tolerant applications in the system. They suggested the simple iterative workload assignment algorithm to run different applications under their Quality of Service requirements. They designed a basic structure based on the virtual machine cloning and synthesis, therefore fully offloaded workload executed in the system according to its characteristics.

Due to the heavyweight burden of virtual machines, many suggestions have been proposing to improve services time of system during workload allocation [6- 8]. The main goal is to reduce services overhead time during workload migration via different optimization methods and framework policies. Through these techniques, many objectives were improved for example, mobile battery consumption, response time, lateness, tardiness and cost of the system. However, still, they only considered delay tolerant applications in their models.
The delay-sensitive and lightweight application-aware container-based frameworks proposed by some researchers [9- 12]. The proposed single runtime environment across the cross-platform, where single executed files of applications can run on different computing nodes. However, different class applications need to run together on a single platform, and those as mentioned earlier, single cross-platform cannot meet these requirements.

The hybrid runtime to run a different class of applications with heterogeneous nodes under resource capacity and deadline constraints has not been formulated yet. This paper devises a novel container-based lightweight hybrid system which can run all class types of applications with lower overhead and less setup time. To ensure the resource capacity and deadline constraints, we plan a new latency aware workload assignment algorithm, which allocates all tasks to the appropriate nodes and migrates workload when it needs.

A) ***Problem Formulation***
The paper considers the A number of applications, i.e., A = {a = 1, . . . , A}. The notation Wa represents workload of specific application a. The request of applications arrive to the system by Poisson process. The study assumes three different kinds of computing nodes for instance, mobile computing, cloudlet computing, and centralized cloud for workload execution i.e., C = {c1, . . . , C}. Whereas Rc={R1, . . . , rR} vector denoted resource capacity of all computing nodes. All computing nodes are distinct by their speeds i.e.,ζr {r1, . . . , rN }. Furthermore, lightweight workload can execute on the mobile device, and delay-sensitive workload can execute on cloudlet, and delay-tolerant tasks can perform on cloud computing.
We formulate this problem assignment problem as follows.

$$x_{W,c} = \begin{cases} 1, \text{ if } x_{xk} = 1 \\ 0, \text{ otherwise} \end{cases} \quad (1)$$

The equation (1) shows binary assignment of workload to the specific computing node during initial assignment, e.g., x = {0, 1}.If it is x{Wa, c} means assignment done otherwise it is zero. We can measure the workload execution into specific node in the following way.

$$T_w^r = \frac{W_a}{\zeta_r}. \quad (2)$$

The equation (2) illustrates the execution of particular workload in the system. However, due to workload migration and preliminary offloading, these applications can incur with communication time, which is showing in the following way.

$$T_w^c = \frac{W_a}{B_w}. \quad (3)$$

The variable Bw demonstrates the available bandwidth of the network during workload migration. The equation (3) shows communication measurement of workload. We denote the response time of each application in the following way.

$$T_{total} = T_w^r + T_w^c. \quad (4)$$

The problem mathematically formulated as follows.

$$\min \sum_{c=1}^{C} \sum_{a=1}^{A} T_{total}. \quad (5)$$

subject to

$$\sum_{a=1}^{A} T_{total} \leq \sum_{a=1}^{APP_N} d_a. \quad (6)$$

da is the deadline of each workload, and equation (6) all workloads must be executed under their deadlines.

$$\sum_{W_a}^{A} W_a \leq \sum_{c=1}^{c} \sum_{r=1}^{R} r. \quad (7)$$

The equation (1) all requested workloads must be less the capacity of computing nodes.

HYBRID COMPUTATIONAL OFFLOADING FRAMEWORK

The proposed hybrid framework consists of three main tiers: Local Thin Client (LTC), Local Cloudlet Server, and Remote Cloud Server, as shown in Figure 1. Initially, users submit applications to the LTC such as APP1, APP2, and APP3. The Local Execution Manager Runtime Environment (Android-X86) run applications files into binaries files. The offloader manager profile applications to understand either it

is the delay-sensitive, the delay-tolerant and the lightweight applications via different profiling technologies. The decision manager based on Network Bandwidth Latency, Application Profiling, Resource Monitor Offline make the decision which application executes where either it is on the local device or to be offload to cloudlet or cloud for execution.

If the decision manager decides delay-sensitive applications, then Cloudlet Execution Manager Runtime Environment executes applications into different phases. The cloudlet server made of Docker engine which creates containers for each application with other binaries and libraries (Lib/ Binaries). The Local Cloudlet Server (LCS) has different modules to ensure the quality of applications. The Resource Monitor Online gives an update of the available capacity of the server. The Execution Time Predictor makes sure the deadline of applications during allocation. The Service Pool and Database checks either requested workloads are capable of allocating or not; otherwise, the Migrator module will offload them to the centralized cloud for execution. However, initial offloading and migration will gain an extra transmission time of application. These runtimes also implemented based on Android X86.

The Remote Cloud Server (RCS) has the same Remote Cloud Execution Manager Runtime Environment (Android-X86) to run either offloaded binaries of files or applications in the system. It is also implemented container Docker technologies, where each container binaries and libraries (Bin/Lib) executes offloaded workload with degrading of QoS of applications. The Service Pool, Resource Monitor, and Bandwidth Monitor are modules to make sure the less communication time, and efficient execution of applications via resource availability.

In Fig 1 local thin client, server and remote server can be seen working on real time mobile cloud APP.

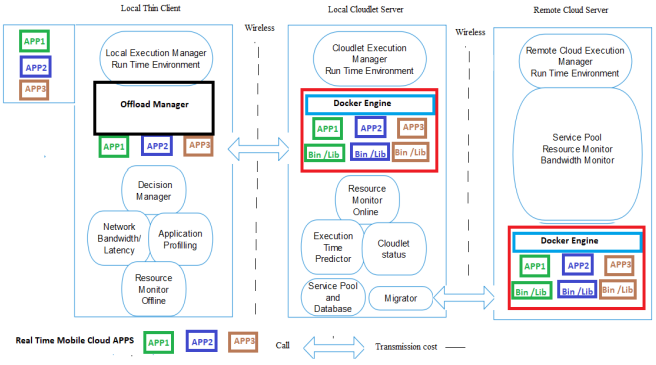

Fig. 1: Hybrid Run Time Framework for Mobile Cloud Application

A) Latency Aware Workload Assignment (LATA)

To cope with the assignment problem, study proposes the Integer Linear Programming (ILP) based iterative optimization Latency Aware Workload Assignment (LATA) method, it determines allocation of all workloads to the different computing nodes. The main goal is to allocated different kind of workloads into different computing nodes in order to minimize response time of applications. We propose a novel LATA algorithm which solves the problem in the following different steps.

Algorithm 1 takes all workloads and nodes and their capacities as input. The goal is to minimize response time of all applications as illustrated in line 2 to 4. The algorithm make initial assignment based on workloads matching to the different nodes as explained in line 5 to 6. It measures the initial assignment of workload with the objective function. After the initial assignment of workloads onto nodes, the deadline and resource capacity must be satisfied as defined from line 8 to 12. If the workload migrates from one node to another, it should satisfies equation (5) with new assignment. Furthermore, if all assignment done, there is no any workload seeks for allocation, then all process of algorithm will close as defined from line 13 to 16.

Algorithm 1: LATA Algorithm

Input : $A = \{a = 1, \ldots, A\}, C = \{c = 1, \ldots, C\}, R = \{r = 1 \ldots, R\}, \sum_{a=1}^{A} d_a$;
Output: $\min T_{total}$;

1 **begin**
2 **foreach** $(a = 1 \quad as \quad A)$ **do**
3 **foreach** $(c = 1 \quad as \quad C)$ **do**
4 The system engine recognizes workload class types as delay-tolerant, lightweight, and delay-sensitive;
5 Make initial assignment based on equation on equation (1);
6 $x = \{a, c\}$=1;
7 Calculate T_{total} based on equation (5);
8 **if** $(a \leftarrow T_{total} \leq d_a)$ **then**
9 Measure the resource capacity based on equation (1);
10 **if** $(W_a \geq c \leftarrow r)$ **then**
11 Migrate workload to another node based on equation (3);
12 Calculate new T_{total} based on equation (5);
13 End of Initial Assignment;
14 The process will continue until all migrated workload must satisfy equation equation (5);
15 End of all assignments;
16 End of main;

IMPLEMENTATION

We have implemented the prototype of the Hybrid proposed framework on Lab server machine. The configuration of each server machine is 3 six core Xeon-IntelX5850, 3.0 GHZ CPU, 20GB RAM (DRAM), 300GB HDD and Ubuntu 16.04. The mobile run time executes on Android devices such as Galaxy Note 8, Galaxy A3, Galaxy S7 and Galaxy J Series and operational with both network base stations (3G/4G) and WIFI connections. Whereas, we evaluated the performance of Cloud based cloudlet run time based on Rattrap environment [12].

A) Experimental Setup

We have two types experimental evaluations such as framework based and algorithm based. However, hybrid framework and proposed algorithm compares with other baselines frameworks and approaches are described follows: Cloud platform run time framework developed in the Android x-86 [5]. Where, every Android x-86 is constructed with 1024 RAM and 2vCPU.

The baseline approaches are Clone Cloud and Cloudlet run time framework designed in the Android x-86 and configured with 512 RAM and 1vCPU.

Conventional approach for computation offloading to the remote cloud with network optimization.

Baseline approaches are earliest deadline first and M/M/1-PS are focused to minimize the execution time of the application at the mobile edge.

The study devises a novel LATA method, which determines the latency efficient allocation of all application onto different nodes. It is an iterative process until all workloads are mapped based on their requirements. LATA is a lightweight process wherein few small iterations all workloads are executing with their properties.

Table1: AliBaba and Amazon Container Based Runtime Resources

Nodes	CORE	MIPS/CORE	Container	Storage(GB)
c_1	i3	1000	2	1000
c_2	i5	3000	4	2000
c_3	i7	5000	8	4000

Table 1 shows the resource implementation in the system based on Alibaba and Amazon. There are three different resources such as c1 is a mobile computing, c2 is a cloudlet computing, and c3 is a centralized cloud. All resources are distinct by their features such as speed, container length, and storage.

B)Different Class TypesWorkload of Applications

The study devises different class types of applications such as LinPack (delay-tolerant), mathematical tool (lightweight), Antivirus (lightweight), 3D Games (delay-sensitive). In Fig 2 we can see the comparison for that we run all applications onto different hybrid system such as baseline approaches[1-5] and Conventional-Approaches [6- 10].

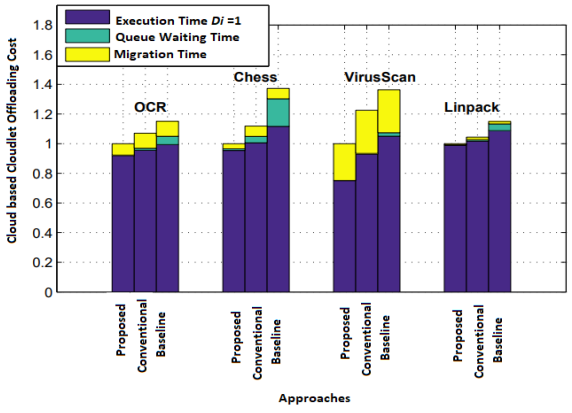

Fig.2. The Performance of Different Workloads in Hybrid Frameworks.

C) Performance Evaluation of different Frameworks and algorithms

This portion discusses the evaluation performances of different class applications onto different runtime environment. The conventional approaches only support mobile and cloud computing delay-tolerant application with the same setting. Whereas, baseline approaches mobile and cloudlet supports delay-sensitive application with a homogenous environment. However, the proposed hybrid framework offers hybrid runtime environment to different class types of applications together with migration techniques. Fig 3 shows LATA outperforms all existing baseline and conventional approaches in term of the response time of applications. LATA schedules all applications with different classes under their deadlines and resource capacity requirements. Hence, it is proved that the proposed hybrid schemes and LATA gaining much more performances as compared to existing studies to run different class types of applications.

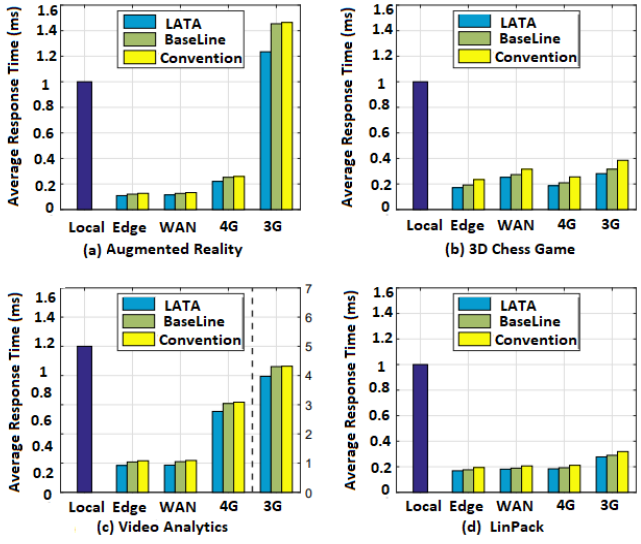

Fig. 3. Performance Analysis of Application Based on Workload

CONCLUSION

This paper presents a novel Mobile Assisted Cloudlet Based Cloud system by using container technology. The main objective of the system is to offer a hybrid runtime environment to run OCR, Augmented Reality, 3D Gaming applications on different computing models. The work devises a novel container technology based hybrid runtime aware framework which divides aforementioned applications into different computing environment in order to minimize response of applications. To cope with the problem, the work suggests a latency aware Task Assignment (LATA) algorithm framework which run aforementioned applications via different steps. The performance evaluation shows that, the proposed system

and algorithm outperform existing monolithic virtual machine systems in term of response time of applications.

In the future work, we will consider the mobility and security in the considered problem during offloading and resource allocation.

REFERENCES

[1] Lei Yang, Bo Liu, Jiannong Cao, Yuvraj Sahni, and Zhenyu Wang. Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds. IEEE Transactions on Services Computing, 2019.

[2] Teh Ying Wah, Ram Gopal Raj, et al. A novel cost-efficient framework for critical heartbeat task scheduling using the internet of medical things in a fog cloud system. Sensors, 20(2):441, 2020.

[3] Abdullah Lakhan, Dileep Kumar Sajnani, Muhammad Tahir, Muhammad Aamir, and Rakhshanda Lodhi. Delay sensitive application partitioning and task scheduling in mobile edge cloud prototyping. In International Conference on 5G for Ubiquitous Connectivity, pages 59–80. Springer, 2018.

[4] Abdullah Lakhan and Xiaoping Li. Content aware task scheduling framework for mobile workflow applications in heterogeneous mobile-edge-cloud paradigms: Catsa framework. In 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pages 242–249. IEEE, 2019.

[5] S. Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In Proceedings of the sixth conference on Computer systems, pages 301–314. ACM, 2011.

[6] Abd El-Hameed G El-Barbary, Layla AA El-Sayed, Hussien H Aly, and Mohamed Nazih El-Derini. A cloudlet architecture using mobile devices. In Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of, pages 1–8. IEEE, 2015.

[7] Abdul Rasheed Mahesar, Abdullah Lakhan, Dileep Kumar Sajnani, and Irfan Ali Jamali. Hybrid delay optimization and workload assignment in mobile edge cloud networks. Open Access Library Journal, 5(9):1–12, 2018.

[8] Hao Qian and Daniel Andresen. Jade: An efficient energy-aware computation offloading system with heterogeneous network interface bonding for ad-hoc networked mobile devices. In Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on, pages 1–8. IEEE, 2014.

[9] Abdullah Lakhan and Xiaoping Li. Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks. Computing, 102(1):105–139, 2020.

[10] Abdullah Lakhan and Xiaoping Li. Mobility and fault aware adaptive task offloading in heterogeneous mobile cloud environments. EAI Endorsed Transactions on Mobile Communications and Applications, 5(16), 2019.

[11] Mian Guo, Quansheng Guan,Weiqi Chen, Fei Ji, and Zhiping Peng. Delay-optimal scheduling of vms in a queueing cloud computing system with heterogeneous workloads. IEEE Transactions on Services Computing, 2019.

[12] Song Wu, Chao Niu, Jia Rao, Hai Jin, and Xiaohai Dai. Container-based cloud platform for mobile computation offloading. In Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International, pages 123–132. IEEE, 2017.