RACE Resource Aware Cost-Efficient Scheduler for Cloud Fog Environment

Sk Mahaboob Basha [1], D Achyuth Kumar [2], K Madhuri [3], Ch Sandeep Sandilya [4], R Siri Chandana [5]

[1] Associate. Professor, Department of Computer Science and engineering

[2,3,4,5] B.Tech., Scholars, Department of Computer Science and engineering

[1,2,3,4,5]QIS College of Engineering & Technology

Abstract- Fog Computing is one of the new computing structures which takes the Cloud to the verge of the network. The structure is formulated for applications that need low latency. Fog Computing has been projected to improve the disadvantages of Cloud Computing. The system is confronted with the variability of dynamic resources that are heterogeneous and distributed. Hence, efficient scheduling and resource allocation are necessary to maximize the use of these resources and the satisfaction of users. In this paper, a resource-aware scheduler RACE (Resource Aware Cost-Efficient Scheduler) is proposed to distribute the incoming application modules to Fog devices that maximize resource utilization at the Fog layer, reduces the monetary cost of using Cloud resources with minimum execution time of applications and minimum bandwidth usage. This RACE comprises of two algorithms. The Module Scheduler in RACE categorizes the incoming application modules according to their computation and bandwidth requirements which are then placed by Compare Module. Comprehensive experimental results obtained from the simulation by using ifogsim simulator show that our approach performs better in most of the cases as compared to the Traditional Cloud placement and the baseline algorithm.

Keywords—Fog Computing, Resource Allocation, Module Schedular, Computing Structures.

I. INTRODUCTION

The advent of IOT has enabled the development of new communities that connects humans and devices. IOT has the potential to improve the quality of life by providing better city management, e-health, and disaster recovery making IoT an attractive platform for public spending. According to D. Manyika et.al. in [1], the theoretical effect of IOT is approximately $11 trillion per annum by 2025. It may reflect 11 percent of the global economy. With the fast growth in the IoT, the number of connected devices has been increased at an enormous pace [2]. In one of the press releases on IOT in 2015 [3], the connected devices to the internet were 13.4 billion, which already outnumbered human beings on the planet. This number will approximately reach 38.5 billion, or five connected devices for every person up to 2020 [3], which is an astonishing figure. This rapid increase in IOT devices led to the generation of a huge amount of data also called Big data. To analyze an enomous amount of information created by a wide range of IOT devices, a lot of computing resources are required. To fulfill the demands of computing resources, a great number of data centers have been set up. The traditional Cloud Computing has been projected to deliver efficient and flexible services to the data service subscribers [4]. The traditional Cloud Computing has two main actors, 1) Cloud users or customers 2) Cloud Service Providers (*CSP*) [5]. CSPs acquire Cloud resources like storage, processing, etc., and make them available to the Cloud users. Cloud users hire the resources which are provided by the CSP on the pay-per-use model [6]. Cloud users access the hired resources through the internet using the CSPs provided interface. Cloud task manager receives the Customer jobs in the form of tasks/Cloudlets and provides these tasks to task scheduling heuristic [7]. This heuristic receives user's task with their computation requirements and assigns to the Virtual Machine(VM) that executes the task with maximum profit. The task scheduling heuristic communicates with the resource manager to acquire VMs related information i.e., VM computation capability and the current workload, etc. Resource manager manages and monitors Cloud resources including both virtual and physical resources.

The shared resources can be dynamically reconfigured and provisioned. Cloud Computing aims to bring together storage and computation in Cloud data centers that connect powerful machines through high bandwidth connections. Cloud Computing offers services to IOT systems at the infrastructure level that can scale storage and processing requirements for IoT devices. However, it is not always the best choice to use Cloud-based computational resources for IOT devices. In most of the cases, the Cloud data centers inadequately fulfill the delay requirements of most of the geo-distributed IOT applications such as health care, emergency response, video streaming, etc. Consequently, delays in-service distribution, congested network, and substandard Quality of Service (QOS) are seen [8].

Data movement from the end-user to the Cloud resources for processing involves delay as it requires high bandwidth for transmission [9]. Cisco in 2012 [10] collaborated with the concept of Fog Computing to solve these problems. Fog Computing is used as an extension of the Cloud Computing model from the core of the network edge [10], [11]. Generally, the Fog Computing environment includes typical network gadgets, like routers, hubs, switches, and other devices. These devices can be located nearer to the IOT devices (presented in figure-1). These gadgets delivered various computing, storage, and network connectivity capabilities and can facilitate the execution of user applications. In addition, Fog Computing facilitates the knowledge of the location, support for mobility, real-time interaction, interoperability, and scalability [12]. The major idea of Fog Computing is putting computing facilities near the data source [13]. Thus Fog Computing works well regarding, network traffic, service latency, and the monetary cost of using Cloud resources, etc. Therefore, Fog Computing can fulfill the IoT applications related to user requirements in a better way better compared to purely relying on Cloud Computing [14].

II. RELATEDWORKS

The resource allocation challenges for user mobility in Fog Computing are highlighted in [19]. In a mobility scenario, when users with different application types (i.e., *latency-tolerant and latency- sensitive*) are in the range of the same device, causes some devices as overloaded (*processor, storage, and/or networking*). To address the mobility issue, Luiz F.et-al., in [19] proposed three algorithms (concurrent, delay priority, and First Come First Served). To reduce the network usage, the concurrent strategy places all incoming application modules to the Fog device in the user's range. The concurrent strategy causes resource contention when there are large numbers of application modules to be scheduled.

First Come First Served *(FCFS)* strategy eliminates the resource contention by placing application modules to the device until the device has much CPU capacity to cope with the incoming modules, while the remaining modules are transferred towards the Cloud without considering the application type (latency-sensitive or latency-tolerant). Although the proposed approach removes the resource contention on the Fog device, yet, it increases the network usage along with delay for latency-sensitive applications.

Considering the application type the delay priority strategy proposed in [19] places the latency-sensitive applications to the nearby devices until devices have enough CPU capacity to accommodate the application modules. Once the Fog device is full (i.e., computation capacity), the remaining modules are mapped on to the Cloud. Although the proposed approach reduces the delay for latency-sensitive applications, yet, with a growth in the number of application modules increases the network usage and monetary cost on Cloud resources.

For effective resource utilization in the Fog Computing environment, a resource-aware scheduling algorithm was proposed in [20]. The resource-aware scheduling algorithm considers CPU, memory, and Bandwidth while assigning application modules to Fog devices. The incoming application modules and devices (Fog and Cloud) are sorted according to their CPU requirement and capacity in ascending order. The selection of the Fog device is based on the recursive binary search until the selected device meets the CPU, Memory, and Bandwidth requirements of the module being assigned. The proposed method does not take into account the cost of Cloud services and also has inefficient use of resources.

A task scheduling algorithm was proposed by Xuan-Qui, et.al. in [21]. The core objectives of this scheduling algorithm are to obtain a balance between the execution time of application and the monetary cost of Cloud resources. The authors have used the Estimated Start Time (*EST*), Estimated Finish Time (*EFT*), and monetary cost of mapping each application module to the device (Fog or Cloud). Modules are assigned to nodes (Fog device or Cloud) based on finding the trade-off value between monetary cost and make span. The proposed approach has poor resource utilization with an increase in the execution time of application.

V.Cardelli et.al in [22] suggested a distributed QoS aware scheduling algorithm for Storm big data processing framework. The Storm is a real-time, open-source computing framework built for dynamically distributed clusters. The authors are expanding Storm with new components to run a distributed scheduler conscious of QoS. To perform experiments the authors have used Apache storm with a cluster of eight worker nodes. The experiments show that the proposed scheduler performs well as compared to the Storm default scheduler with improvement in application execution performance. Jianhua Fan, et.al. in [23] presented a deadline aware task scheduling framework in tiered based IoT infrastructure. The primary emphasis is to optimize the service provider 's benefit by scheduling and positioning modules.

The problem was introduced as a problem with the 0-1 knapsack. To get the Fog service provider with full cumulative net income, an optimization algorithm focused on Ant Colony Optimization (ACO) is implemented. The findings of the solution presented would be compared with FCFS and Min-min algorithms. Resource utilization and latency have not been evaluated in this study. Jinlai Xu, et.al. [24] proposed a centralized edge computing resource allocation method, called Zenith.

The resource allocation method allows establishing a contract between service-provider and edge infrastructure providers. These contracts allow edge servers to deliver efficiently their assigned computing resources and schedule the application modules offloaded in a way that minimizes the latency. The authors proposed a decoupled resource distribution model. The proposed model handles the distribution of computing resources allocated at the edges independently of the facilities rendered by the service provider. The proposed approach does not consider the monetary cost of using Cloud resources.

Disadvantages of Existing system are:

- In the existing work, Fog scheduler are not reduced makespan, not minimized monetary cost, no higher throughput, reduced energy Consumption, etc.
- First Served (FCFS) strategy is not eliminate the resource contention by placing application modules to the device until the device has much CPU capacity to cope with the incoming modules.

III. PROPOSED SYSTEM ARCHITECTURE

In the proposed system, a resource-aware scheduler is presented to distribute the incoming application modules to Fog devices. The placement strategy of Application modules from Fog layer to the Cloud layer introduced in our approach reduces the monetary cost of using Cloud resources with minimum execution time and minimum bandwidth usage. The main contribution of this paper are as follows:

A Resource Aware Cost Efficient scheduler is proposed for the placement of application modules from Fog layer to the Cloud layer instead of directly sending it to the Cloud in order to achieve maximum resource utilization.

A priority mechanism is defined for the placement of application modules to minimize the bandwidth, execution time and monetary cost of using cloud resources.

We have employed Resource Aware policy in iFogSim-simulated fog environment and compared with tradition cloud and Cloud Fog placement approaches from different standpoints. The performance results show significant improvement in favor of our policy

Advantages of Proposed System

- The system designs a fast access management with RACE (Resource Aware Cost-Efficient Scheduler).
- The system uses in which Scheduling is the process of mapping application modules onto computing resources to efficiently utilize computing resources (such as CPU, memory, bandwidth, etc.,).

IV. RESULTS AND DISCUSSION

The results obtained after executing the implementation code is shown from Fig.1 to Fig.9.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

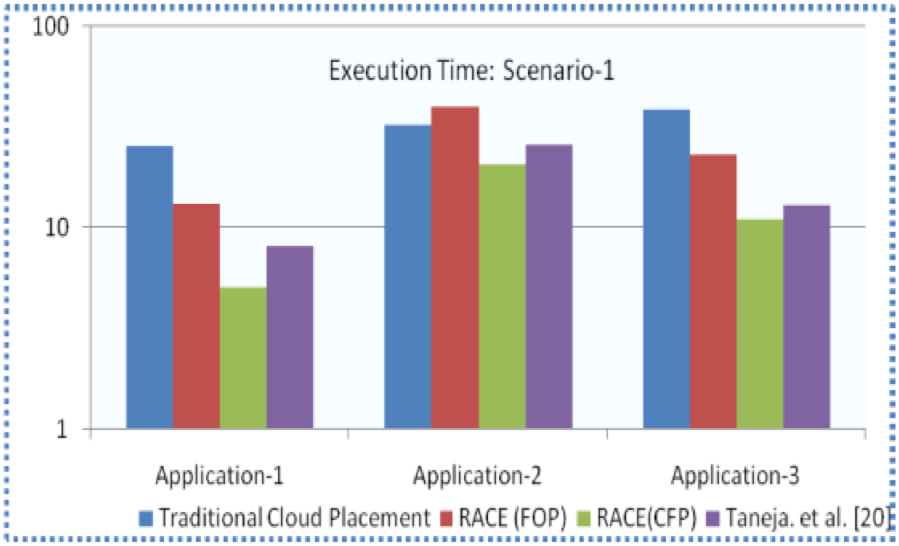

Fig.1 Execution time of three different applications with 4-FD

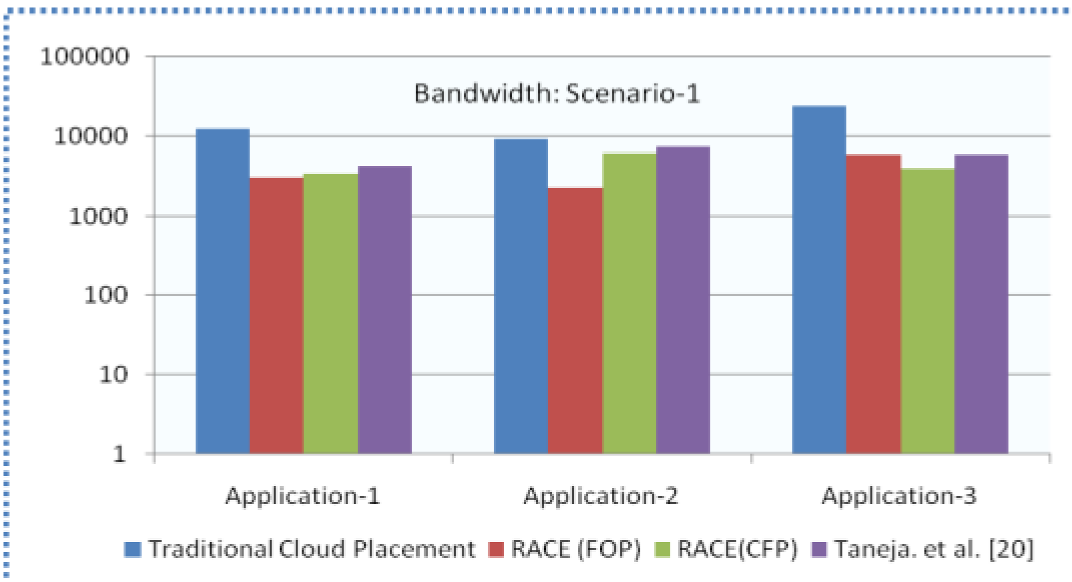

Fig.2 Bandwidth consumption of three different applications with 4-FD.

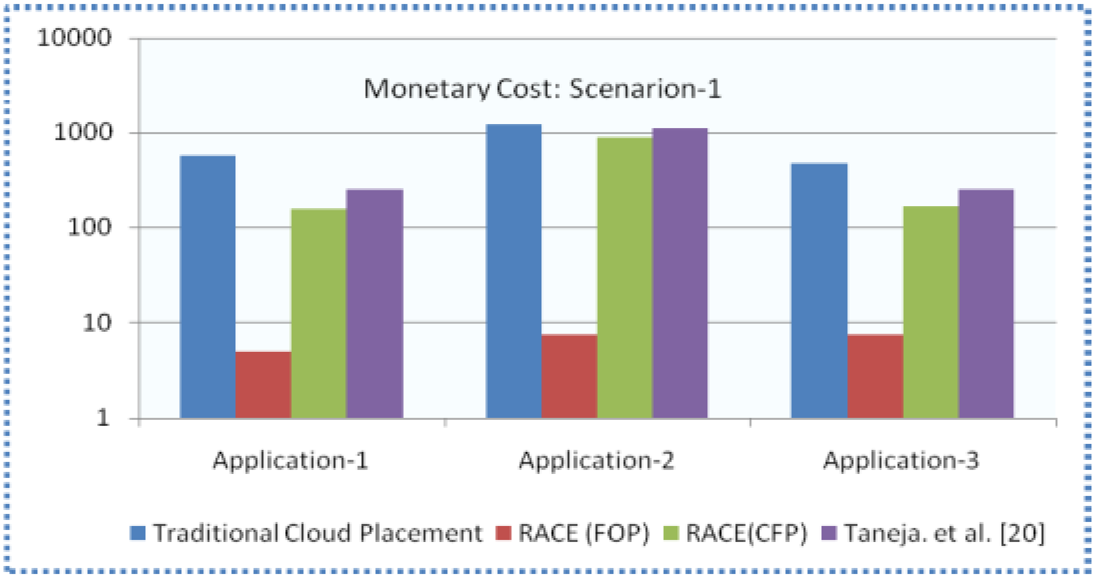

Fig.3 Monetary cost of three different applications with 4-FD.

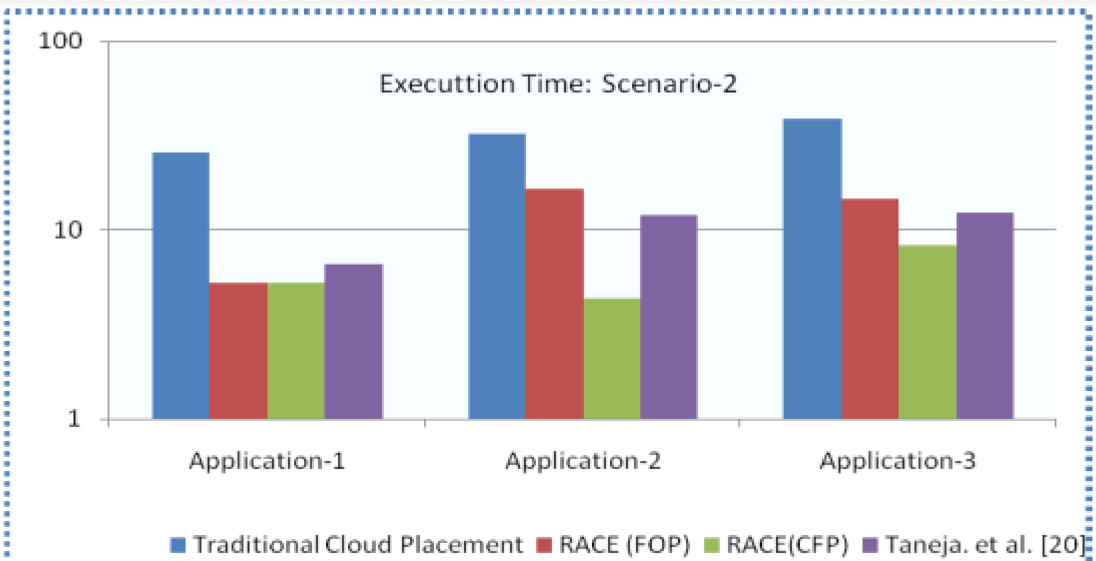

Fig.4 Execution time of three different applications with 8-FD.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

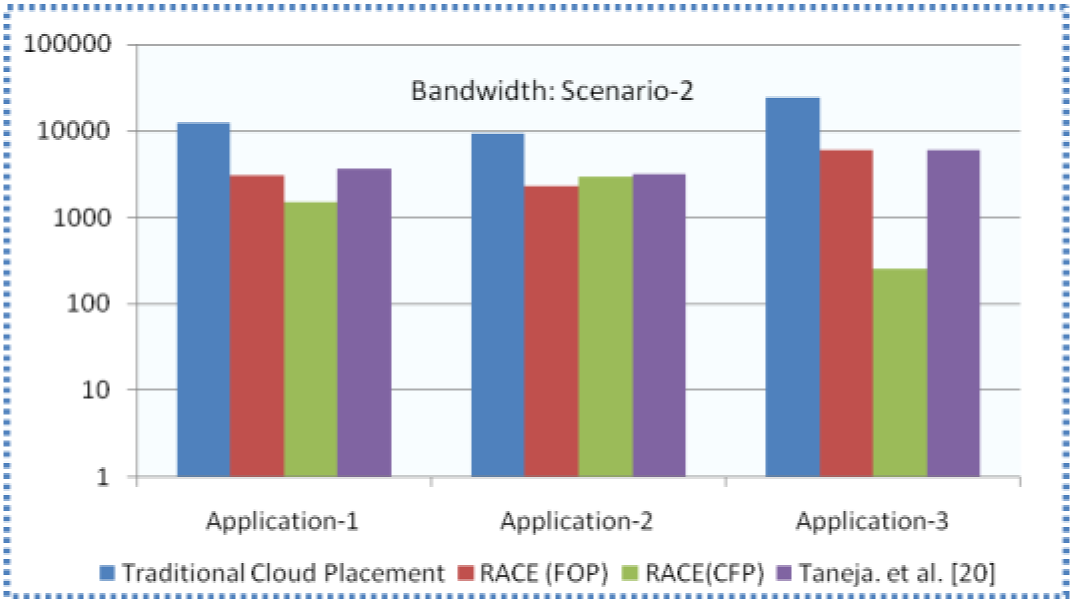

Fig.5 Bandwidth consumption of three different applications with 8-FD.

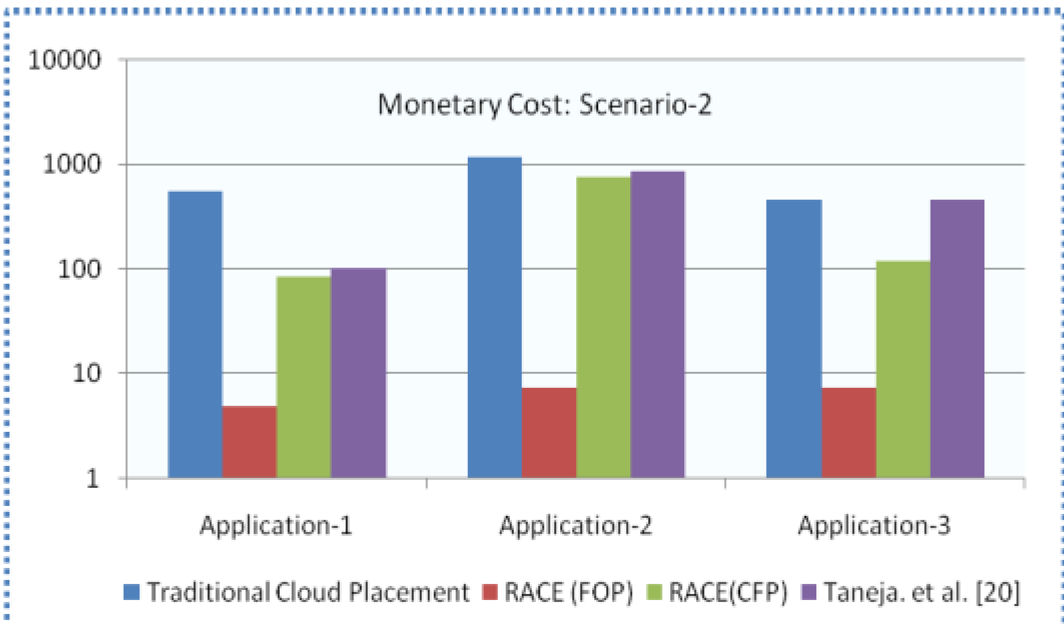

Fig.6 Monetary cost of three different applications with 8-FD.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

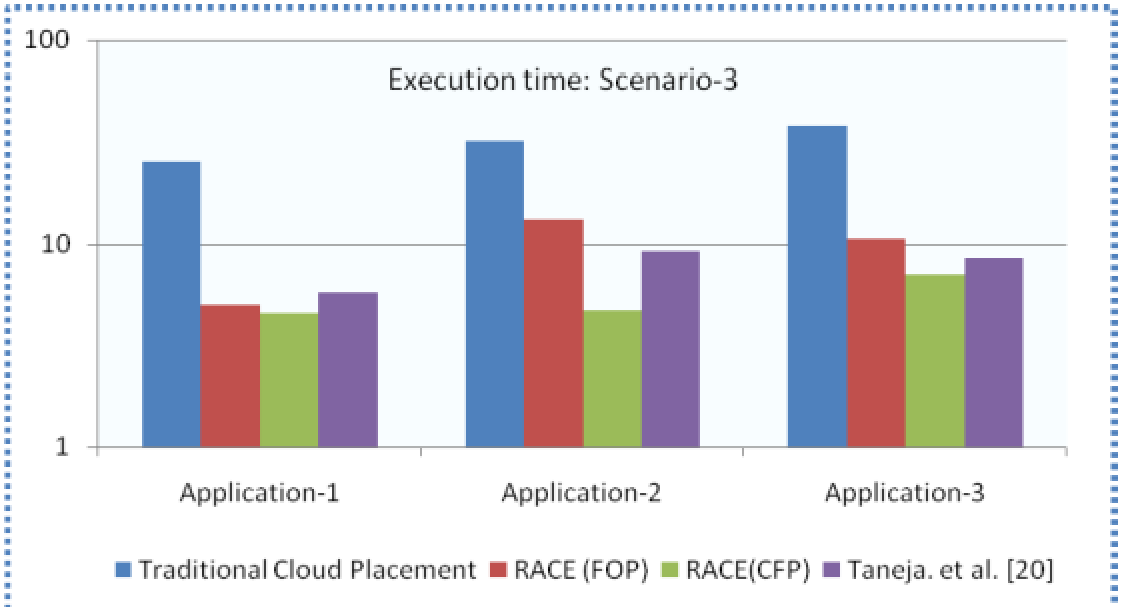

Fig.7 Execution time of three different applications with 10-FD.

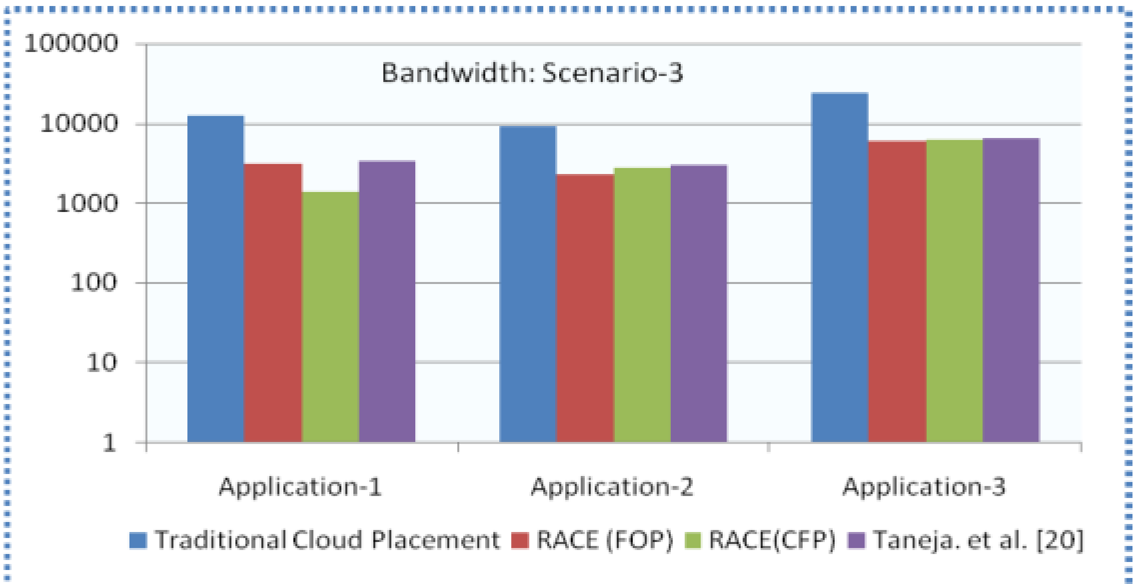

Fig.8 Bandwidth consumption for three different applications with 10-FD.

ISSN- 2394-5125 VOL 9, ISSUE 04, 2022

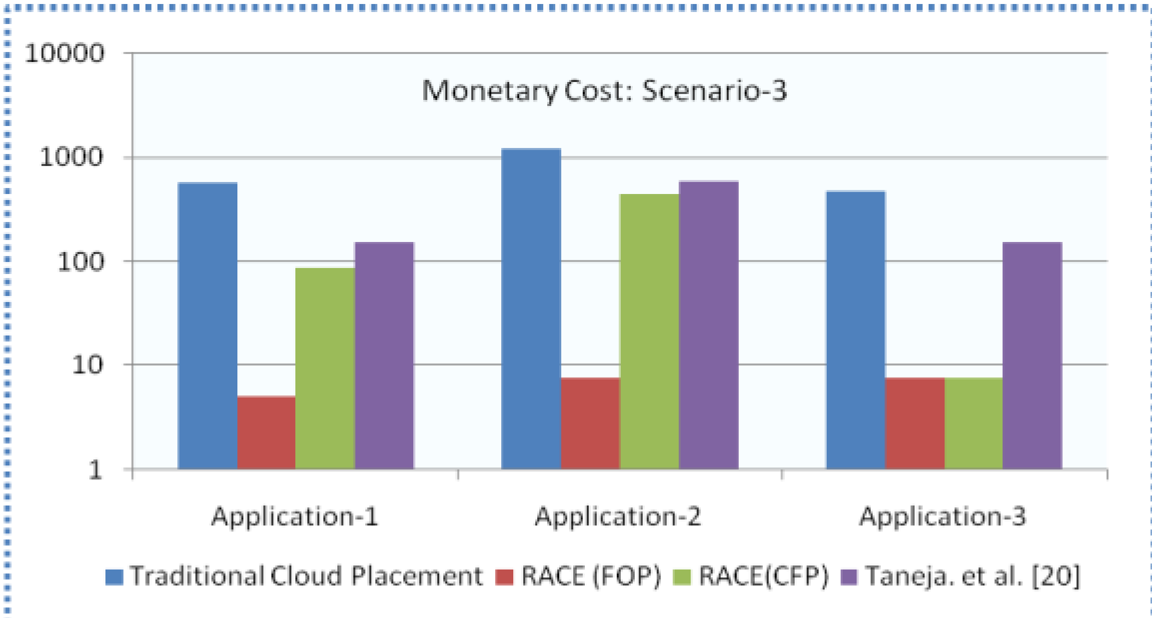

Fig.9 Monetary cost of three different applications with 10-FD.

V. FUTURE SCOPE AND CONCLUSION

The Fog Computing model has a tremendous ability to help a broad range of IOT applications. In this paper, we presented a Resource Aware Scheduler (RACE) for the placement of application modules in three-tier Fog Cloud architecture. The workload is defined in terms of incoming applications with their modules. We have tested the RACE algorithm by using three different strategies i.e., placement of the workload directly on the Cloud, placement of workload on the Fog layer and the placement of workload from Fog layer to the cloud layer. The RACE(CFP) performs in favorable ways for all workloads in terms of minimum execution time, lowest bandwidth, and least monetary cost. Although the RACE(FOP) performs better in some cases for bandwidth consumption and monetary cost when number of fog devices are greater. However the overall performance for the execution time, bandwidth consumption and monetary cost the RACE(CFP) have n better results.

REFERENCES

[1] J. Manyika *et al.*, “McKinsey Global Institute: Unlocking the potential of the Internet of Things,” *McKinsey Glob. Institute.*, pp. 1–144, 2015.

[2] F. S. Collins, E. D. Green, A. E. Guttmacher, and M. S. Guyer, “A vision for the future of genomics research,” *Nature*, vol. 422, no. 6934, pp. 835–847, 2003, doi: 10.1038/nature01626.

[3] S. (2015). Smith, “‘Internet of Things’ Connected Devices to Almost Triple to Over 38 Billion Units by 2020 - Juniper,” *http://www.juniperresearch.com/press/press- releases/iot-connected-devices-to-triple-to-38-bn-by-2020*, 2016.

[4] B. P. Rimal, E. Choi, and I. Lumb, “A taxonomy and survey of Cloud Computing systems,” in *INC, IMS and IDC, 2009. NCM’09. Fifth International Joint Conference on*, 2009, pp. 44–51.

[5] M. Nabi, S., Ahmed, "OG-RADL: Overall Performance Based Resource Aware Dynamic Load-balancer for Deadline Constrained Cloud Tasks," *J. Supercomput.*, 2020.

[6] M. Ibrahim *et al.*, "An In-Depth Empirical Investigation of State-of-the-Art Scheduling Approaches for Cloud Computing," *IEEE Access*, vol. 8, pp. 128282–128294, 2020, doi: 10.1109/ACCESS.2020.3007201.

[7] A. Hussain, M. Aleem, A. Khan, M. A. Iqbal, and M. A. Islam, "RALBA: a computation-aware load balancing scheduler for Cloud Computing," *Cluster Comput.*, vol. 21, no. 3, pp. 1667–1680, 2018, doi: 10.1007/s10586-018-2414-6.

[8] S. Sarkar and S. Misra, "Theoretical modelling of Fog Computing: A green computing paradigm to support IoT applications," *IET Networks*, vol. 5, no. 2, pp. 23–29, 2016, doi: 10.1049/iet-net.2015.0034.

[9] Sandvine, "Sandvine Global Internet Phenomena Report 2H-2015," *Sandvine Corp.*, p. 17, 2015, doi: 10.1007/s13398-014-0173-7.2.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and its role in the internet of things," *Proc. first Ed. MCC Work. Mob. cloud Comput. - MCC '12*, no. August 2012, p. 13, 2012, doi: 10.1145/2342509.2342513.

[11] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Chapter 4 – Fog Computing: principles, architectures, and applications," *Internet of Things*, pp. 61–75, 2016, doi: 10.1016/B978-0-12-805395-9.00004-6.

[12] R. Lavanya, "Fog Computing and Its Role in the Internet of Things," *Adv. Consum. Fog Comput. Archit.*, p. 63, 2018.

[13] J. Cao, Q. Zhang, and W. Shi, "Challenges and Opportunities in Edge Computing," in *Edge Computing: A Primer*, Springer, 2018, pp. 59–70.

[14] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, 2018, doi: 10.1109/TCC.2015.2485206.

[15] A. T. Thien and R. Colomo-Palacios, "A Systematic Literature Review of Fog Computing," *Nokobit*, vol. 24, no. 1, pp. 28–30, 2016.

[16] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on Fog Computing: architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, 2017, doi: 10.1016/j.jnca.2017.09.002.

[17] M. Hajibaba and S. Gorgin, "A review on modern distributed computing paradigms: Cloud Computing, jungle computing and Fog Computing," *J. Comput. Inf. Technol.*, vol. 22, no. 2, pp. 69–84, 2014, doi: 10.2498/cit.1002381.

[18] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and applications," in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 2015, pp. 73–78.

[19] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-Aware Application Scheduling in Fog Computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, 2017, doi: 10.1109/MCC.2017.27.

[20] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, 2017, pp. 1222–1228.

[21] X. Q. Pham and E. N. Huh, "Towards task scheduling in a cloud-Fog Computing system," *18th Asia-Pacific Netw. Oper. Manag. Symp. APNOMS 2016 Manag. Softwarized Infrastruct. - Proc.*, 2016, doi: 10.1109/APNOMS.2016.7737240.

[22] V. Cardellini, V. Grassi, F. Lo Presti, and M. Nardelli, "On QoS-aware scheduling of data stream applications over Fog Computing infrastructures," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*, 2015, pp. 271–276.

[23] J. Fan, X. Wei, T. Wang, T. Lan, and S. Subramaniam, "Deadline-aware task scheduling in a tiered IoT infrastructure," in *GLOBECOM 2017- 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[24] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-Aware Resource Allocation for Edge Computing," *Proc. - 2017 IEEE 1st Int. Conf. Edge Comput. EDGE 2017*, pp. 47–54, 2017, doi: 10.1109/IEEE.EDGE.2017.15.

[25] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in Fog Computing based on priced timed petri nets," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1216–1228, 2017.