International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-9, Issue-1, 2022 Regular Edition E-ISSN: 2349-7084

Priority Based Task Scheduling and Delay Optimization in Mobile Edge Computing

R. Yamuna[1], M. Usha Rani[2]

[1]Research Scholar, [2]Professor, Dept. of Computer Science, SPMVV, Tirupati.
Email ID: ryamunaspmvv@gmail.com , mur@spmvv.ac.in

*Corresponding Author: ryamunaspmvv@gmail.com

Available online at: http://www.ijcert.org

Received: 29/12/2021, Revised:15/01/2022, Accepted: 21/01/2022, Published: 27/01/2022

Abstract: - Day by day the numbers of Internet of Everything (IoE) devices are increasing which produce massive amounts of data every day. Cloud computing handles such massive amount of data. Cloud computing is a model that provides on-demand computing, storage, and network resources with little or no interaction from service providers. A challenging issue in the cloud is resource scheduling and delay optimization to enhance cloud service providers' profits by ensuring the quality of services (QoS) demanded by users. Particularly in smart health care the response time plays an important role. In this paper, a task scheduling algorithm is proposed which assigns the resources based on the priority. The requests are classified into three categories highly delay sensitive, moderate delay sensitive and low delay sensitive based on the attribute values like blood pressure, heart rate and temperature. The execution time is then optimized by setting a threshold value in order to provide services with less delay. The overall performance is increased by 40.1% compared to other scheduling methods.

Keywords: Smart healthcare, delay, Task scheduling, priority, IoT.

1. Introduction

The Internet of Things (IoT) is a broad concept that encompasses a wide range of day to day objects such as home appliances, computer devices, animals, farms, industries, and cars, all of which are linked together via heterogeneous networks. IoT has made these objects ‘‘smart," allowing them to perceive, process, and communicate effectively over a network to conduct important tasks without the need for end-user interaction.The initial goal of the Internet of Things (IoT) was to embed intelligence only in physical objects, but Cisco later improved the concept to create the Internet of Everything (IoE), which focuses on connecting people, data, processes and things to build intelligent relationships.

Smart agriculture, healthcare monitoring, traffic, and other IoE systems use the Cloud-centric Internet of Things (CIoT) architecture for processing, storage and analytics. CIoT data centre’s are typically located multiple hops away from a source node which causes long delays.

Pervasive computing and the development of 4G/5G technologies have also brought computing services everywhere by linking billions of new IoT apps and devices.It results in the generation of a massive amount of data which is unable to handle by Cloud effectively, resulting in network congestion and high latency. Smart home automation and other delay-sensitive, real-time, and geo-spatially dispersed IoT applications are still in development.Cloud computing cannot effectively service applications that demand low latency, such as healthcare monitoring, smart traffic surveillance, virtual reality, and other applications.

Several techniques, such as Fog Computing,

 https://doi.org/10.22362/ijcert/2022/v9/i01/v9i0101

Mobile Computing, Edge Computing, and others, have been proposed to overcome these CIoT constraints, it was to provide networking capabilities, decision-making, storage, and processing in close proximity to the end-user application.

Mobile Edge Computing (MEC) is a new computing model that addresses issues like latency and resource utilization. Mobile Edge Computing is a novel processing paradigm that fulfils the ever-increasing computing demands of mobile applications.MEC's main feature is that it transfers mobile computing, network control, and storage from resource-constrained mobile devices to network edges, enabling applications that demand a lot of compute and a lot of latency to run.

MEC enables a mobile application to track real-time data such as behavior, location, and environment, minimizing the exchange of sensitive data between the mobile device and cloud resources while also conserving energy. Edge resources, on the other hand, are constrained in terms of compute capability, scalability, and energy sensitivity, making them inappropriate for long-term processing.

Integrating edge and cloud resources is one approach for availing advantage of cloud resources' capabilities and the availability of access resources at the edge layer. Mobile Edge Cloud Computing (MECC) is the term referring to such a kind of joint computing architecture..

2. Related Work:

The performance of mobile/wireless networks has a huge impact.to the responsiveness and processing speed of mobile applications [1]. According to [2,] a huge percentage of mobile users would prefer to execute mobile applications locally due to network performance concerns. It's hard to decide whether to execute an application locally or remotely, and it requires continuous network monitoring and application profiling [3].

In [4] adopted a game-theoretic method to compute offloading in MECC, separating the execution of multi-user applications using the waiting time in compute nodes. These researchers offered effective methods for implementing cloudlet/edge computing to improve mobile app responsiveness, minimize latency, and reduce device energy consumption. They did not, however, integrate MCC and MEC, nor did they propose practical and acceptable resource allocation and task scheduling solutions for data-intensive mobile applications. Furthermore, the proposed solutions have little to do with enhancing offloading decision behavior based on data-aware factors as data size and application location.

[5] Proposed a threshold-based strategy for improving MEC QoS by integrating local edge resources with public cloud resources, combining the low latency of local clouds with the vast computing capabilities of public clouds at the same time.

For multi-site compute offloading in MECC, [6] designed heuristic approach. Multiple objective optimizations of time, money, and energy were evaluated in this work. The goal was to use a heuristic strategy to convert a multi-weight optimization to a single-weight optimization.

[7] examined at resource allocation in fog computing by looking at several types of applications and using three different scheduling policies: FCFS, concurrent, and delay-priority. Video surveillance (VSOT) and the EEG tractor beam game are two applications that incorporate the algorithms (EEGTBG). The first is a near-real-time application, and the second is a delay-tolerant application.[8] proposed a prioritized job scheduling methodology in fog computing to reduce overall response time and cost. When a job arrives, its priority is determined by its deadline. The computed priorities of a work are used to determine where it should be placed in the Fog layer.

There are several micro data centre's and Fog nodes in each Fog layer that can communicate with one another. The workload is transferred to Cloud if all of the data centres in a Fog layer are saturated. Other significant goals, such as energy consumption and network utilization, are ignored.In a Cloud-Fog scenario, [9] suggested a genetic algorithm for job scheduling optimization. Their goal was to reduce the amount of time it took for jobs to be executed. Each chromosome indicates a node's task assignment. To create a new population, mutation and crossover were used.

In their work on Real-Time Task Assignment (RTTA), [10] employed deep reinforcement learning and evolutionary approaches to schedule real-time jobs and use a neural network trained by reinforcement learning.

[11] Proposed an approach for categorizing tasks into categories or classes based on their qualities, with each class including tasks with similar characteristics. The tasks inside each class are scheduled in order of execution time, and the classes are scheduled depending on the weights supplied to characteristics.

When the existing allotted resources for the virtual machine are insufficient, [12] established a resource allocation strategy for dynamic placement of VM on Host. The strategy incorporates placement of Virtual Machine for Applications at the initial deployment of these apps onto the cloud.

In [13] data uploading, slicing, indexing, encryption, dissemination, decryption, retrieval, and merging are all part of the proposed framework. The hybrid encryption technique was created to ensure the security of huge data prior to its storage in many clouds. Real-time cloud storage environments are used in the simulation analysis.

For breast health monitoring, a suggested IoT-cloud-based health care (ICHC) system framework [14] has been developed. Using this approach, the earliest probable breast cancer signs can be identified.

In [15] proposed a new technique for gathering additional real-time traffic data that is based on a genuine cloud model. Developed a hybrid approach dubbed smart road traffic information service (RTIS) that combines WSN and VANETs to transform traditional transportation into an intelligent transportation system

.3. Proposed Work:

The architecture is shown in the figure1. It has mainly three layers. User layer, Edge layer and Cloud layer.

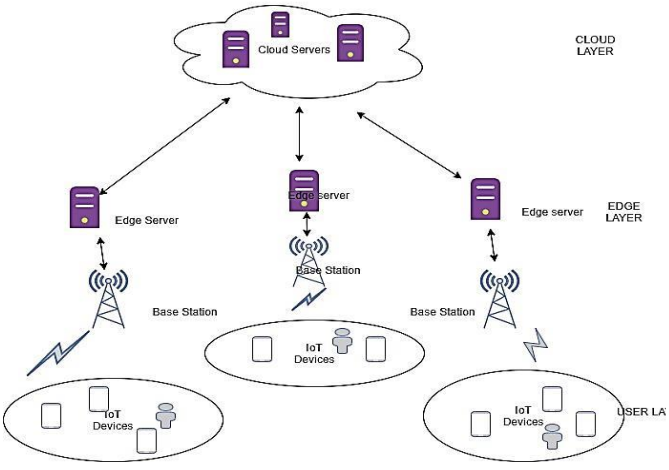

Figure1. Mobile Edge Architecture

User layer: Sensors and actuators make up the IoE layer. Cameras, temperature, heartbeat sensors, GPS sensors, humidity sensors, and other sensors obtain raw data from the outside environment, convert it to signals, and send it to Fog nodes for processing. Fog nodes deliver the results of their processing to actuators, which act as controllers and take appropriate action.

Edge layer:Between the Cloud and the end devices, the fog layer serves as an intermediary. It strongly combines the Cloud and IoE layers of the Fog, enabling both independent evolution and high levels of interaction between them. The Fog layer serves as a backbone to the core IoE platform when using the Cloud services mentioned above.

The Edge layer, which is made up of heterogeneous Fog devices with limited compute, storage, and networking capabilities, consists of routers, switches, proxy servers, and cellular base stations. Edge servers with resources including processors, RAM, and storage that are connected to edge devices. A Cloud server is connected to these servers.

Cloud layer:Cloud is the architecture's topmost layer. It collects information from networking devices for long-term activity and data analysis, and then sends the results to Edge devices for action.

In e-healthcare, cloud-based solutions result in increased latency, which is undesired in emergency situations. With Edge computing, a significant number of healthcare computing jobs can be completed by local Edge servers, resulting in lower latency and increased availability.In smart health care applications, there are various use cases, some of which are critical and others which are delay-tolerant. For example, a doctor's data is recorded and saved so that he or she can review it later. Delays are tolerated in some cases.While faster data processing is necessary in some cases, such as when a patient is in a critical condition, it is not always necessary to generate emergency alerts. Because a delayed reaction to an emergency notification can threaten the patient's life, such tasks have a higher latency requirement.

In the proposed work, the requests are classified into three types.

1. High delay sensitive.
2. Moderate delay sensitive.
3. Low delay sensitive.

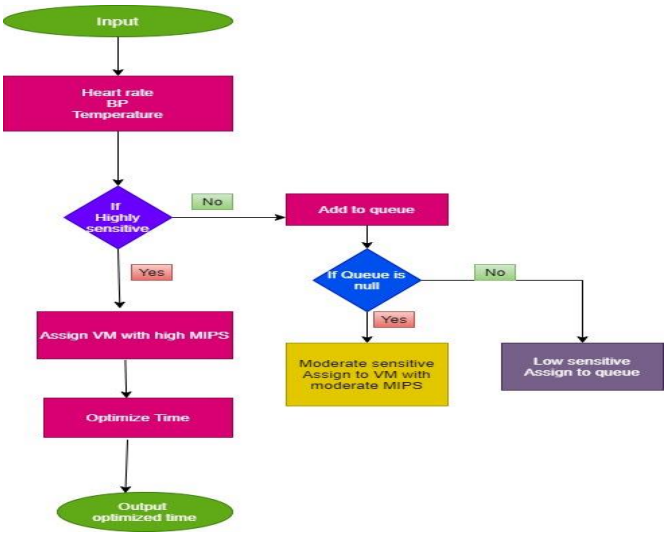

Figure2. Proposed Architecture

In the first case few attributes were considered. They are high blood pressure, temperature, colestral and heart rate. In the second case, the values of these attributes are moderate and used only for booking an appointment. Hence considered as moderate delay sensitive. In the third case, the values of these attributes are almost normal and seek less medical attention. Hence considered as low delay sensitive.

The input is collected and given to the edge server. The server then analyses the severity based on the attribute values.

1. If the request comes under emergency i.e., highly delay sensitive, then it will be assigned to the cloud server with high MIPS computation capability. Thus the response will be given in short time.
2. If the request comes under moderate delay sensitive, then the request will be assigned to the server immediately which is having moderate computation capacity only if the waiting queue is empty.
3. If the request comes under low delay sensitive, then it will be assigned to the waiting queue. It will be given low priority.

The time taken to process the request is calculated using number of instructions, computation capability of the server, data size and data rate.

$$Tt = \frac{CPU\ cycles}{Computation\ capacity} + \frac{Data\ size}{Data\ rate} \quad (1)$$

$$T_t = \frac{Ii}{CCi} + \frac{Dsi}{Dri} \quad (2)$$

Where,

$T_t$is the Total time taken.

I_i is the number of CPU cycles.

CC_i is the Computation capability.

$D_{si}$is the Data size of the task.

D_{ri} is the Data rate.

To optimize the total time a threshold value has been taken. It is represented as α. The value is taken from the mean of all generated values.

The optimized value can be given as,

$$T_o = (1-\alpha)T_t \quad (3)$$

4. Experimental evaluation and Results

The algorithm is implemented in Cloud Simulator which provides a virtual environment where we can deploy infrastructure. The results are presented in the tables and graphs. The obtained results are compared with existing methods FCFS (First Come First Serve) and SJF (Shortest Job First).

The algorithm is implemented by varying number of devices. Initially 100 devices were taken and calculated the time. Then the algorithm is repeated with 200, 300, 400 and 500 devices and assigned the resources accordingly.

Table1. Time taken for different nodes

Number of Nodes	**Time taken(msec)**	
	SJF	**Proposed**
100	200	170

200	230	220
300	245	230
400	250	235
500	257	242

Table2. Time with Optimization

Number of Nodes	Time taken(msec)	
	Obtained	Optimized
100	170	125
200	220	130
300	230	183
400	235	190.5
500	242	210

Table1 presents the time taken to process the request for existing algorithm and proposed algorithm. For 100 nodes, in SJT it is taking 200 msec and for proposed the time taken has been reduced to 170 msec. Similarly, for other nodes also the response time has minimized in proposed algorithm compared to SJF algorithm.

Table2 presents the response time using proposed algorithm before applying optimization and after applying optimization. For 100 nodes initially it is taking 170 msec and after applying optimization the value has been reduced to 125 msec. Similarly, for other nodes.

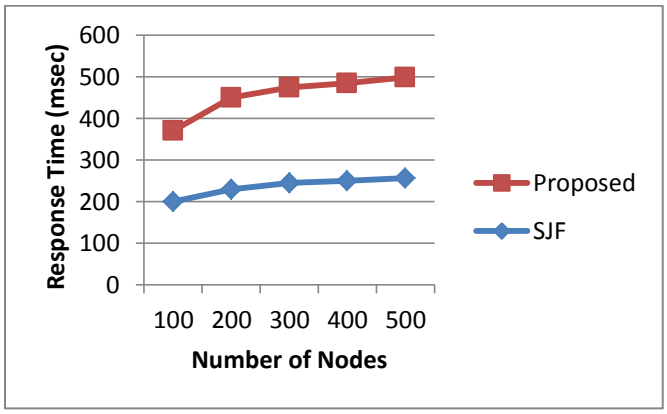

Figure 3. Response time by varying nodes

From the results, it is observed that the proposed method is taking less time to allocate resources and overall performance is improved by 40.1%.

5. Conclusion:

The number of Internet of Everything (IoE) devices is growing every day and generating vast amount of data. Cloud computing is capable handles that data whichallows users to access computing, storage, and network resources on demand. Few applications like smart healthcare are delay sensitive. A task scheduling method is proposed that assigns resources based on priority. Based on attribute values such as blood pressure, heart rate, and temperature, the requests are divided into three categories: highly delay sensitive, moderate delay sensitive, and low delay sensitive. In order to deliver services with less delay, the execution time is then minimized by choosing a threshold value. The overall performance is increased by 40.1%.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al., A view of cloud computing. Commun. ACM 53 (4), 50–58, 2010.

[2] Hung, S.-H., Shih, C.-S., Shieh, J.-P., Lee, C.-P., Huang, Y.-H.,. Executing mobileapplications on the cloud: framework and issues. Comput. Math. Appl. 63 (2),573–587, 2012.

[3] Giurgiu, I., Riva, O., Juric, D., Krivulev, I., Alonso, G.,. Calling the cloud: enablingmobile phones as interfaces to cloud applications. In: Proceedings of the

10th ACM/IFIP/USENIX International Conference on Middleware. Springer-Verlag NewYork, Inc., p. 5, 2009.

[4] Goudarzi, M., Zamani,Cardellini, V., Person, V.D.N., Di Valerio, V., Facchinei, F., Grassi, V., Presti, F.L.,Piccialli, V.. A game-theoretic approach to computation offloading in mobilecloud computing. Math. Program. 157 (2), 421–449, 2016.

[5] Zhou, B., Dastjerdi, A.V., Calheiros, R., Srirama, S., Buyya, R., b. mcloud: acontext-aware offloading framework for heterogeneous mobile cloud. IEEE Trans.Serv. Comput. 10 (5), 797–810, 2015.

[6] Enzai, N.I.M., Tang, M.,. A heuristic algorithm for multi-site computationoffloading in mobile cloud computing. ProcediaComput. Sci. 80, 1232–1241, 2016.

[7] Choudhari T, Moh M, Moh T-S. Prioritized task scheduling in fog computing. In: Proceedings of the ACMSE 2018 Conference; 2018; Richmond, KY.

[8] Bittencourt LF, Diaz-Montes J, Buyya R, Rana OF, Parashar M. Mobility-aware application scheduling in fog computing. *IEEE Cloud Comput*.;4(2):26-35, 2017.

[9] Mishra S, Jain S. Ontologies as a semantic model in IoT. *Int J Comput Appl*. 2018. https://doi.org/10.1080/1206212X.2018.1504461

[10] Nguyen BM, ThiThanhBinh H, Do Son B. Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application incloud–fog computing environment. *Applied Sciences*.;9(9):1730, 2019.

[11] Mai L, Dao N-N, Park M. Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latencyminimization in fog computing. *Sensors*.;18(9):2830, 2018.

[12] M. Shelar, S. Sane, V. Kharat, and R. Jadhav, "Autonomic and energy-aware resource allocation for efficient management of cloud data centre," in 2017 Innovations in Power and Advanced Computing Technologies (*i*-PACT), pp. 1-8, IEEE, 2017.

[13] Viswanath, G., and P. Venkata Krishna. "Hybrid encryption framework for securing big data storage in multi-cloud environment." *Evolutionary Intelligence* (2020): 1-8.

[14] Kavitha, Modepalli, and P. Venkata Krishna. "IoT-Cloud-Based Health Care System Framework to Detect Breast Abnormality." In *Emerging Research in Data Engineering Systems and Computer Communications*, pp. 615-625. Springer, Singapore, 2020.

[15] Kavitha, S., and P. Venkata Krishna. "Realistic Sensor-Cloud Architecture-Based Traffic Data Dissemination in Novel Road Traffic Information System." In *Emerging Research in Data Engineering Systems and Computer Communications*, pp. 639-653. Springer, Singapore, 2020.

