Available online at
www.mdl.mazedan.com

MJCEA

www.mazedan.com

www.mazedan.com/mjcea

AN INTERNET OF THINGS ENVIRONMENT, A CIVIL ENGINEERING CUSTOMIZED APPLICATION DEVELOPMENT EMPLOYING DYNAMIC DATA SCHEDULING

MAZEDAN JOURNAL OF CIVIL ENGINEERING & ARCHITECTURE

e-ISSN:

Article id- MJCEA0103004
Vol.-1, Issue-4
Received: 18 Nov 2021
Revised: 16 Dec 2021
Accepted: 18 Dec 2021

ZAINEB NAAZ*, VIDUSHI SHARMA

Citation: Naaz, Z., & Sharma, V. (2021). An Internet of Things environment, a civil engineering customized application development employing Dynamic Data Scheduling. *Mazedan Journal of Civil Engineering & Architecture*, 1(4), 27-31.

Abstract

With the rising proliferation of Big Data and the Internet of Things, edge computing serves as an efficient computing model for time-stringent data processing which can bypass network bandwidth and latency limitations, and has become one of the cores of interconnected devices. The billions of IoT devices are producing huge and voluminous data of different characteristics. It may be critical, periodic, or priority-based data. However, the solution to schedule data based on its characteristics is yet to be defined. Traditional scheduling algorithms fall short in satisfying the needs of an IoT network that has a truly global scale. In this paper, based on the concept of edge computing and job scheduling algorithms we have proposed a dynamic scheduling algorithm that will schedule data based on its characteristics. We highlight the distinctive parameters of data that we believe should be addressed in an IoT network. We finally compared our dynamic scheduling technique with the traditional round-robin scheduling technique. The main aim of our proposed model is to minimize the delay of critical data that needs immediate action. Experiment evaluation results demonstrate that dynamic scheduling of data reduces the response delay of data that needs immediate action.

Keywords: Cloud Computing, Dynamic Scheduling, Edge Computing, Internet of Things (IoT), Round Robin Scheduling

1. INTRODUCTION

With the evolution of various network technologies, Internet of things is one of the most emerging and extremely attractive technology that is making inroads into the field of communication. Worldwide availability of internet, unprecedented fall in cost of sensors, and highly integrated high-performance SoC (System on Chip) computing solutions are key driving forces behind massive growth of IoT. Internet of Things has applications in various domain such as health care, industries etc. Several applications of Internet of things are: Smart grid, Smart home, Smart Farming, Wearables, Smart Energy Management, Smart Industries. There are several advantages of IoT such as: (i) IoT minimize the human efforts and allow things to interact with each other without human intervention, (ii) Internet of things improves security by making a system efficient, (iii) IoT saves human time, (iv) With the help of internet of things we can utilize our resources efficiently. In a smart home each and every physical thing is connected to the internet with the help of a gateway and perform their task without human intervention. The owner of the smart home can see the working of the smart things through the app or a website. Adding more smart things in the home tends to generate more data in smart home environment. Also, critical data will also be there in the smart home which needs immediate attention. Managing the data from various smart things is a tremendous task. Therefore, an efficient algorithm is needed for managing the various types of data in an IoT environment like in a smart home. In this paper we have proposed an algorithm for managing various types of data in an Internet of Things environment.

In today's era smart things being ubiquitously deployed and we have an ocean of data in our real-life environment. In Internet of Things environment, the speed of data growth has far outperformed the load limitations of network bandwidth. Cloud Computing [1] has high computational power and storage space, though some defects exist in a centralized data processing mode. The traditional model of centralized data processing uploads the enormous data collected from device to the central cloud for unified processing, and then sends processed results to the terminals. Despite the fact of the emergence of interconnected things [2], bottleneck and delay of network bandwidth are outlined [3]. For this concern, in recent years, edge computing [4, 5] has been introduced as a supplement to cloud computing [6], where industrial IoT applications should be handled as much as possible in a distributed and localized fashion [7]. It is noteworthy that the processing of sensory data queries is an essential ingredient of traditional industrial IoT applications [8]. Considering the functional diversity of smart things and the nature of future events to be examined, a dynamic scheduling algorithm has been proposed in this paper.

To evaluate the efficiency and applicability of our proposed technique extensive experiments are conducted.

School of Information and Communication Technology, Gautam Buddha University, Greater Noida, India
***Corresponding author email:** zainebnaaz119@gmail.com

The outcomes show that this strategy performs better in IoT environment.

The rest of the paper is organized as follows. Section 2 introduces relevant concepts. Section 3 introduces the Proposed System Model for IoT. In section 4 our proposed approach implementation and evaluation has been presented. In section 5 our proposed work is concluded.

2. PRELIMINARIES: CONCEPTS

In this section the relevant concepts and literature of importance are presented:

Edge Computing

Edge computing is a model in which we run data analytics at network's edge where the data is generated because cloud computing is not always effective when the data is produced at the edge of the network [4].

Concept Definition

Internet of Things is basically an internetwork of physical objects which are embedded with sensors, computers, connectivity, and actuators. With the help of sensors, actuators and connectivity these physical objects acquire data, transform it into knowledge, make intelligent decisions and generate physical actions to manipulate the environment. With the help of sensors, we can control and monitor day to day life equipment's in IoT. Sensors collect data from their environment. In general, the data collected by sensors are analog in nature, these sensors convert raw analog data into digital signals and transmit them to its control center, around 60-70% of IoT data computations run on the cloud. In this way, any changes in the environment can be monitored remotely via the internet from anywhere [9].

For Internet of Things services, our main focus must be on harnessing the hidden potential of data. IoT devices produce heterogeneous and geographically-dispersed real-time data and periodically report the occurrence of normal or abnormal events of interest. The most demanding observations are periodic ones in terms of communication overhead and storage due to their streaming and continuous nature, while events present time-strain with end-to-end response times depending on the urgency of the response required for the event [10]. Hence, scheduling data based on its urgency and criticality is a matter of concern. Scheduling of jobs in IoT mainly focuses on executing the jobs to complete a particular job or response. There are many scheduling algorithms that the scheduler follows like FCFS, SJF, Round Robin, Priority etc. Each scheduling scheme has some drawbacks. In FCFS the jobs with high priority or less burst time have to wait till their turn. In SJF scheduling, also the jobs with high priority or high burst time have to wait till the jobs with low burst time get served. Here also the high-priority jobs have to starve and this algorithm is not for practical use. In priority scheduling, if the job with high priority is considered first but other jobs have to face starvation. In priority scheduling, if the job with high priority continues to come again and again will never get a chance to be executed. In Round Robin scheduling a time slice is set. Each job is allotted only that quantum of time for execution. Thus, each job has to wait a long time to get executed. Here, the average waiting time and the average turnaround time is very high [11].

To address the challenge of executing a task based on its priority, we have proposed a dynamic scheduling algorithm, which will make a queue of data based on its urgency and will schedule the data according to that queue.

Performance Parameters

The three 3 essential parameters have been taken for dynamic scheduling of data/processes at cloud/network's edge in the internet of things environment. These performance parameters are as follows:

Critical Data: These are the values that express or involve an analysis of the faults of a process. These types of values require an immediate response.

Prioritized Data: These are the values that are assumed as the fact or condition of being regarded or treated as more important than others.

Periodic Values or Normal data: These are low priority values that can wait for response or action.

Since traditional scheduling algorithm does not work in the internet of things environment because the data is voluminous, the network is data-centric, data is both events based as well as periodic data and response time of server should depend on the type of data which includes critical data, priority-based data, and periodic data. Critical data is the data that has to be processed first as they depict critical values (the value of data that needs immediate response). These critical data originate in the health care monitoring system. After going through several research papers on the internet of things, cloud computing, edge computing a dynamic algorithm is proposed, according to this dynamic scheduling algorithm, a queue will be formed of data such as critical data will be followed by priority data followed by normal data. Based on this critical data will be entertained first as this data can't wait if we want our IoT environment efficient. So, the purpose is that before sending data on the server, data should be arranged based on its classification and after this, it will be scheduled based on the round-robin algorithm, the reason for using the round-robin algorithm is that it guarantees no starvation. The purpose of this dynamic algorithm is that critical data should be processed first then priority data and later on normal data.

3. PROPOSED SYSTEM MODEL

A working model of IoT based system is proposed using the concept of edge computing and job scheduling algorithms. The system performs computation on data at network's edge before sending it to the cloud. Figure 1 shows the proposed system model.

The dynamic scheduling algorithm is proposed according to the system model, and it will be implemented at the edge of the cloud i.e., Edge computing will be done before cloud computing. The following algorithm will analyze the data based on critical value, priority values, and periodic. If processes have critical value, then the process having the highest critical value will be served first then based on priority processes will be arranged on the edge of the cloud. Then in cloud computing, Round Robin

Algorithm will be used. The reason for using Round Robin is that there is a guarantee of No Starvation in Round Robin. The proposed algorithm for dynamic scheduling in the internet of things environment is given in table 1.

Table 1 Dynamic Scheduling Algorithm

```
node* inputNodes []; node *head =
NULL; node *tail = NULL;
 while (! inputNodes.Empty())
{
node * newNode = createDoublyLinkedListNode(inputNodes[i]); if (head == NULL)
 {
 head = newNode; tail = head;
}
else {
node *traverseBack = tail;
 while(traverseBack->criticality > newNode->criticality) { traverseBack =
   traverseBack -> previous;
}
    while(traverseBack->criticality == newNode->criticality && traverseBack->priority > newNode -> priority)
{
traverseBack = traverseBack -> previous;
}
insert newNode after traverseBack
 }
 }
```

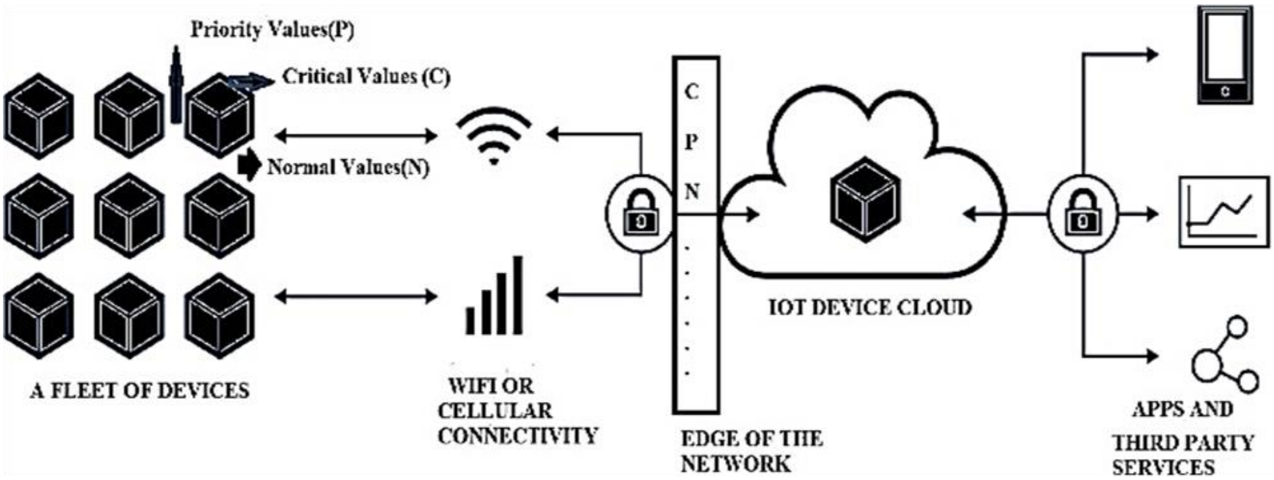

Figure 1 Proposed System Model for IoT

The proposed algorithm will arrange the data in a queue according to its characteristics such that if a critical event occurs the critical data will be first in the queue, then priority-event data will follow the critical data and then normal data will be last in the queue. So, if the data coming from devices has high critical value than priority value then it would be processed first. Then data with a high priority will be processed. Later on, normal data will be processed.

4. IMPLEMENTATION AND EVALUATION

The above-proposed algorithm for dynamic scheduling in the internet of things was simulated using the Java Eclipse IDE. We have considered the case when we have normal data along with critical data as well as priority-based data. Different process names along with the burst time priority and criticality are given in table 4. Burst time denotes the time required for the job to complete its processing. Arrival time denotes the time when the particular job arrives in the sequence. For the given set of data, the sequence for execution for each process according to the proposed algorithm is given in the following table 3.

Using the proposed dynamic scheduling algorithm, a queue of data has been formed, where the Critical data is being followed by priority data which is further being followed by the normal data.

Evaluation Results

The working of the proposed algorithm has been analysed by executing the data set assumed in table 2 using a simple round-robin algorithm based on the arrival time of processes and by applying our proposed dynamic algorithm on the data at the network's edge and then execute the dynamically arranged data by using round-robin scheduling algorithm.

Applying the dynamic scheduling and simple round-robin algorithm on the data we have taken in the sequence of execution and service time are given in the following table 4.

Table 2 Process id, burst time, arrival time, priority, criticality

Process Name	Burst Time	Arrival Time	Criticality	Priority
A	3	0	0	0
B	5	0	0	0
C	2	0	0	0
D	3	0	0	9
E	4	0	4	8
F	5	0	0	7
G	6	0	0	11
H	2	0	6	0
I	4	0	7	0
J	5	0	0	12
K	6	0	5	0
L	4	0	0	10

Table 3 Process id, burst time, arrival time, priority, criticality

Process Name	Burst Time	Arrival Time	Criticality	Priority
I	4	0	7	0
H	2	0	6	0
K	6	0	12	0
J	5	0	0	12
G	6	0	0	11
L	4	0	0	10
D	3	0	0	9
E	4	0	0	8
F	5	0	0	7
C	2	0	0	6
A	3	0	0	5
B	5	0	0	4

Table 4 Service Sequence data

Algorithms	Service Sequence	W.T of Most Critical Data	T.T of Most Critical Data
Dynamic Scheduling	IHKJGLDEFCABIKJ GLDEFABKJGFB	26ms	22ms
Round Robin	ABCDEFGHIJKL ABDEFGIJKLBFGJK	32ms	36ms

Comparison of algorithms for waiting time

In figure 2 a comparative analysis of both techniques is given according to the data given in table 2 based on the waiting time of each process. We have calculated the waiting time and turnaround time of all processes and a comparison has been done.

The chart shows that process I which has high critical value has a waiting time of 32ms in a simple round-robin technique and 22ms in a dynamic scheduling technique. Hence, we can say that the waiting time of a critical process has been reduced.

Comparison of algorithms for a total delay

In figure 3 a comparative analysis of both the techniques is given according to the data given in 2 based on the total delay of critical processes. Here, a total delay means the complete execution time of a process.

In figure 3 the graph shows that process, I which has a high critical value has a total delay of 36ms in a simple round-robin technique but after applying dynamic scheduling at the edge of the network the total delay has been reduced to 26ms.

5. CONCLUDING REMARKS

Edge computing has enormous potential for handling millions of heterogeneous sensors and networks under the Internet of things great evolution. An algorithm is proposed to perform edge computing at the edge of the network so that we can schedule data dynamically.

The waiting time and total delay of both the scheduling techniques compassion shows that. the waiting time and total delay in the dynamic scheduling technique are lesser than the traditional round-robin scheduling technique. In short, the problem of scheduling data based on its characteristics has been solved.

REFERENCES

[1] Bahrami, Mehdi, and Mukesh Singhal. "The role of cloud computing architecture in big data." Information granularity, big data, and computational intelligence. Springer, Cham, 2015. 275-295.

[2] Culler, David E. "The once and future Internet of everything."GetMobile: Mobile Computing and Communications 20.3 (2017): 5-11.

[3] Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., "A view of cloud computing." International Journal of Computers and Technology 4, 2013, 50–58.

[4] Shi, Weisong, et al. "Edge computing: Vision and challenges. "IEEE internet of things journal 3.5 (2016): 637-646.

[5] Liu, Yang, et al. "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach." Computer Network 129 (2017): 399-409.

[6] C. Zhu, H. Zhou, V.C.M. Leung, K. Wang, Y. Zhang, L.T. Yang, toward big data in green city, IEEE Commun. Mag. 55 (11) (2017) 14–18.

[7] K. Kaur, S. Garg, G.S. Aujla, N. Kumar, J.J.P.C. Rodrigues, M. Guizani, Edge com- puting in the industrial internet of things environment: software-defined-net- works-based edge-cloud interplay, IEEE Commun. Mag. 56 (2) (2018) 44–51.

[8] G. Cong, C.S. Jensen, D. Wu, Efficient retrieval of the top-k most relevant spatial web objects, in: Proceedings of the VLDB Endowment, 2009, pp. 337–348.

[9] P. Suresh, J. Vijay Daniel, V. Parthasarathy, R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment." International Conference on Science Engineering and Management Research (ICSEMR), 2014.

[10] Mervat Abu-Elkheir, Mohammad Hayajneh, and Najah Abu Ali; "Data management for the internet of things: design primitives and solution" Sensors, 2013. 13(11):15582-15612.

[11] CPU Scheduling in Operating Systems, URL: https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems.