Advances and Applications in Mathematical Sciences
Volume 20, Issue 12, 2021, Pages 3194-3204

ORGANIZING IoT HEALTH CARE BASED ON FOG COMPUTING: CORONA TASKS SCHEDULING BASED ON THE PRIORITY

KESHETTI SREEKALA and B. INDIRA

Department of Computer Science and Engineering
Mahatma Gandhi Institute of Technology
Gandipet, Hyderabad-500075, India
E-mail: ksrikala_csa@mgit.ac.in

Department of MCA
Chaitanya Bharathi Institute of Technology
Gandipet, Hyderabad-500075, India
E-mail: bindira_mca@cbit.ac.in

Abstract

IoT in HealthCare is an important and emerging area as Mobile health and monitoring patients remotely is possible with the current available technologies. At present worldwide the CORONA patients and CORONA death rate is increasing and there is no place for patients in the hospitals. This is the actual time to utilize IoT HealthCare services and since it is related to health no delays are encouraged which may cause due to transmission of data to the cloud and then back to the application. Hence we come up with a new solution which introduces fog computing in between sensor and cloud computing so that much of the processing is done in fog layer which can reduce the amount of data to be transmitted between sensor and cloud and reduce the delay in transmission of data that improves the performance of IoT HealthCare System. Irrespective of the amount of data generated we need to schedule these tasks based on the priority that is severity of the disease.

Introduction

IoT is defined as the computational devices that are interconnected through the Internet and are embedded in our activities that are able to send and receive the data [1]. IoT is opening doors to the intercommunication among things and individuals and among things. IoT stood as the entrance

2010 Mathematics Subject Classification: 68.

Keywords: Cloud Computing, Fog Computing, IoT, Task and Scheduling.

Received November 16, 2020; Accepted December 15, 2020

gate for humans to lead a comfortable life by providing different Facilities like remote health care, use of ICT to increase operational efficiency in smart cities, quick response, Industrial IoT, smart agriculture and so on. IoT applications are dependent on the services provided by cloud computing. The cloud provides on demand services such as computing resources, data storage and processing capabilities that are needed by IoT. Such an important IoT is facing certain constraints like low bandwidth, congested network, fault tolerance support, privacy and security of data. HealthCare and monitoring corona patients' kind of applications which are dependent on IoT need quick response and analysis of real time data. This kind of applications cannot tolerate delays caused by sending the data to the cloud, processing the data and then returning the results from the cloud to the application. In order to reduce the delays we use another layer called fog computing layer that sits between cloud computing layer and the IoT layer. In this paradigm, the fog computing layer is located nearer to the end users and acts as a local cloud for the processes that need to work on real time data and quick response. This set up allows IoT applications on the edge of the connected devices instead of cloud computing layer [2]. Fog computing brings large number of advantages to IoT applications like increased performance, real time data processing with minimal delays and quick response, greater efficacy in handing over real time and sensitive data to the clients, handling data center failures, provision of reciprocative services, support of mobility, auto deployment, dynamic configuration, reduction of data transfer between cloud and IoT devices, reduction of network traffic and so on [3].

With the above mentioned advantages scheduling of Corona tasks based on fog computing can play vital role as this is the major health industry monitoring problem. Based on the severity of the disease we give priority to the patients. This makes us to redesign fog scheduling algorithm such that it is compatible with the health care industry. Decision making is very important as most of the Corona symptom patients are under home quarantines or government provided quarantine centers.

The main objective of this study is to improve the corona patient monitoring remotely on time and lessen the death rate of the patients and at the same time minimizing rate of doctors affected with Corona. To achieve this we use a classified task scheduling method based on priority. A set of

rules, procedures and policies are used to schedule tasks in the fog computing on to the available resources. This is called Task Scheduling.

The following sections of this paper are organized as indicated below. In the following Sections we describe literature survey, Architecture of IoT HealthCare and monitoring with the use of Fog Computing, motivations, proposed scheduling method, simulation tool and the set up, then the results and finally we write the conclusion.

Literature Review

As fog computing is meant for increased performance scheduling of tasks in fog computing is attracting many researchers. The following are some of the observations make by us. Lindong Liu et al. [4] proposed. "A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment". This strategy minimized total execution time and total waiting time. Swati Agarwal et al. [5] proposed. "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing". They proposed a model which can handle resource allocation, resource overflow and fault tolerance problems. In this model there are three layers cloud layer, fog layer and client layer. By using this algorithm cost of data transfer time and response time are decreased where as utilization of bandwidth is increased but when the required resource is not available in the fog layer it sends the task to cloud computing layer. Xuan-Qui Pham et al. [6] proposed. "A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing". They could achieve a balance between performance of the application and charges needed for utilizing cloud services. Thus the authors say that they have achieved greater performance at lower cost. L. Georgios et al. [7] proposed. "A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments". This model uses three layers and schedule minimized communication and high computation oriented tasks in cloud computing layer where as maximized and low computation oriented tasks in the fog computing layer. This approach also considers the cost of data movement between application layer and fog layer. The authors said that this approach gives 79.69% lower dead line miss ratio when compared to the baseline policy. These are motivations of our proposed work.

Fog Computing Based Architecture for IoT HealthCare

Fog computing based architecture for IoT health care is of three layers. They are sensors layer, fog computing layer and the central cloud computing layer. This kind of architecture is shown in figure 1.

The Sensors layer or the devices layer

The devices that are present in the sensors layer are meant for collecting the corona patient data or a quarantine centered person data. All the data that is collected in this lower layer is sent to the fog computing layer with the help of Wifi or any cellular network. These devices are attached to the body of the patient or quarantined person.

Fog computing layer

In this layer the actual sensed data of the patients is processed. Plasma cytokines and chemokines are measured periodically and they are compared with the values of a healthy person which are stored in a database. Since HealthCare system is operated in real time environment it quickly responds to the patient about the disease and stores the data in the cloud computing layer. That means the fog computing layer is acting as a bridge between cloud computing and service layers.

Cloud computing layer

The cloud computing layer is the actual center where each patient's data [8] is permanently stored and also if there is larger computation or analysis is required it is processed in this layer. In future if the patient's data need to be studied by the doctor fog computing layer retrieves all the data from the cloud layer. Thus cloud computing layer acts as a central storage of the Health Care system.

Motivations

Task Scheduling algorithms assign priority to the task based on the length of the Queue. But this is not fair as always the lengthy tasks that are patient suffering with Corona from a longer period of time will be waiting in the Queue and Smaller tasks that are most recently Corona detected cases will be served first. This situation leads to Shortest Job First. There is

another variation where lengthy and medium size jobs are given high priority and smaller tasks will be waiting that is person suffering from Corona with maximum number of days or sufficiently good number of days will get treatment first whereas recently Corona detected patients have to wait for the treatment. This resembles max min algorithm.

If assigning the Corona Patients to each of the available system sequentially irrespective of the length of the jobs to be processed at that system, wherever the greater length Queue is there delay causes in response. This is intolerable as the Corona Patient must get the treatment immediately. This motivated us to work toward minimizing the length of the Queue, waiting time and execution time. Irrespective of the length of the Queue, The treatment to the patient may be very important. Hence there should be a way to assign priority to any of the patients/tasks. This motivated us to do this work. The above stated motivations are very important to us as they motivated us to work toward developing a new scheduling algorithm.

Proposed Scheduling Algorithm

Corona Task Scheduling in IoT Health Care is very much needed at present situation. The classified task scheduling method based on priority overcomes the limitations that are addressed in section IV. This technique divides virtual machines into different categories and the CORONA Tasks are classified based on the priority of the task. Virtual machines are divided into different categories depending on the experience/expertise of Doctor who is taking care of that virtual machine tasks. Next Corona Patient tasks are divided into different groups based on the priority of task and these groups are mapped to the suitable virtual machine.

Corona tasks can be classified into 6 different classes based on the day of attack.

Day1 Tasks: Patients run with fever. Muscle pain, Dry cough and fever are symptoms from day 1 to day 4.

Day 5. Tasks: Patients experience difficulty in breathing from day 5 to day 6.

Day 7. Tasks: This is the day most of the patients would think to consult

doctor for treatment.

Day 8. Tasks: The patients develop acute respiratory distress syndrome and severity of the disease is bad. Lungs are filled by the fluid.

Day 10. Tasks: The patients will develop abdominal pain, loss of appetite sometimes they may die also. Patients are generally treated on ICU in this stage.

Day 17. Tasks: On average people will get discharged from the hospital.

Corona Task scheduling by using a classified task scheduling method based on Priority.

This method works based on the priority of corona task instead of the length of the task. Classified task scheduling method based on priority operated in two stages.

Stage 1. Depending on the severity of the disease we divide the Corona cases into three different categories. The three categories are Critical cases, medium cases and start/end cases. These three classes are defined below.

Critical cases: Day 8 and Day 10 tasks fall under this category.

Medium cases: Day 5 and Day7 tasks fall under this category.

Start/End cases: Day 1 and Day 17 cases fall under this category.

Stage 2. Virtual Machines are divided into 3 categories based on the experience of the doctor using it.

Category 1 VM: Doctors with experience and experts in Corona monitoring will be mapped to these machines.

Category 2 VM: Doctors with moderate experience will be mapped to these machines.

Category 3 VM: Junior doctors are mapped to these machines.

Classified task scheduling method based on priority is carried out as stated below.

When the tasks arrive they will be classified into three cases depending on the priority of the case. This depends on the severity of the disease. These three types of cases are mapped on to three different categories of virtual

machines depending on priority/experience of the doctor. Highest priority task will be mapped to expert doctor's VM.

Flow Chart of CORONA Task Scheduling based on priority in OT HealthCare

The following figure 2 shows the flow chart that demonstrates how the classified task scheduling method works based on the priority of tasks in IoT HealthCare which is based on Fog Computing

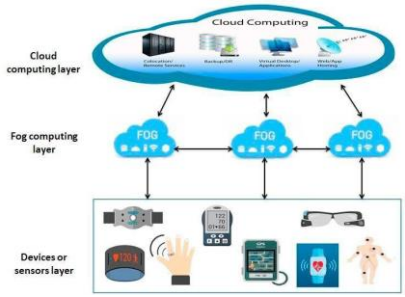

Figure 1. Fog based architecture for IoT.

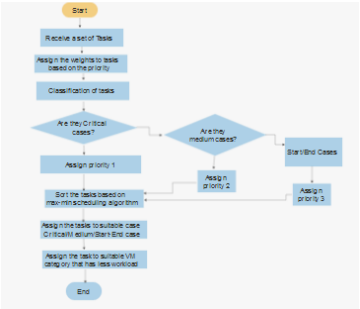

Figure 2. Classified task scheduling Health Care based on priority of tasks.

Impact of Classified task scheduling based on priority over max-min scheduling algorithm

The data that is available is first tested using max-min scheduling algorithm and then it is measured using the classified task scheduling based on priority, both the methods are compared to find the performance achieved by this proposed method. The performance of the system will be measured in terms of Total waiting time (TWT), Total Execution time (TET) and Total Turnaround time (TTT). The relative priority of each task is set by the length

of the task.

Max-Min functionality

The Max-Min algorithm without classified task scheduling sorts the tasks from greatest to smallest based on the execution time. Then these tasks are mapped on to virtual machines starting from the greatest execution time. As a result tasks with less execution time need to wait irrespective of their importance. Table 2 depicts the scenario of max min scheduling algorithm and the related tasks.

Table 1. List of Task length, Task Priority and Task ID.

Length of the task	Priority of the task	Task Identity
T=3000	Low priority	T2
T=6000		T5
T=2000		T1
T=10000		T9
T=13000	Medium priority	T12
T=11000		T10
T=70000		T6
T=40000		T3
T=114000	High Priority	T13
T=500000		T4
T=800000		T7
T=900000		T8
T=112000		T11

Table 2. The Max-Min scheduling without classification of tasks.

Tasks	T8	T7	T4	T13	T11	T6	T3	T12	T10	T9	T5	T2	T1
length	900000	800000	500000	114000	1120000	70000	40000	13000	11000	10000	6000	3000	2000
VM	3	2	1	3	2	1	3	2	1	3	2	1	3
TWT	0	0	0	360	320	370	420	360	440	450	370	475	456
TET	360	320	370	60	40	70	30	10	35	6	5	8	1
TTT	360	320	370	420	360	440	450	370	475	456	375	483	457

Classified task scheduling based on priority. Classified task scheduling method works based on the priority of corona task instead of the length of the task. Depending on the severity of the disease we divide the Corona cases into different categories called Critical cases, medium cases and start/end cases. Virtual Machines are also divided into 3 categories based on the experience of the doctor using it. Doctors with experience and expertise in Corona monitoring will be mapped to Category 1 VMs, Doctors with moderate experience will be mapped to category 2 VMs and junior doctors are mapped to category 3 VMs.

When the tasks arrive they will be classified into three cases depending on the priority of the case. This depends on the severity of the disease. These three types of cases are mapped on to three different categories of virtual machines depending on priority/experience of the doctor. Highest priority task will be mapped to expert doctor's VM. There can be more than one VM

in each category but for simplicity and easy to understand the process we are considering one virtual machine in each category. Table 3 shows the mapping of tasks and their turnaround time.

Table 3. Classified task scheduling based on priority.

	High priority tasks					Medium priority tasks				Low priority tasks			
Tasks	T8	T7	T4	T13	T11	T6	T3	T12	T10	T9	T5	T2	T1
length	900000	800000	500000	114000	1120000	70000	40000	13000	11000	10000	6000	3000	2000
VM	1	1	1	1	1	2	2	2	2	3	3	3	3
TWT	0	360	680	1050	1410	0	70	100	110	0	6	11	19
TET	360	320	370	60	40	70	30	10	35	6	5	8	1
TTT	360	680	1050	1110	1150	70	100	110	145	6	11	19	20

Table 4. Comparisons of Max-Min scheduling and classified task scheduling based on priority.

	Max-Min	Classified Task Scheduling based on priority
ATWT	309.3	293.5
ATET	101.15	101.15
ATTT	410.46	371.61

From Table 3 it is clear that starving of medium priority jobs and low priority jobs is greatly reduced.

Table 4 depicts comparison between Average TWT, Average TET and Average TTT.

Table 4 shows that in classified task scheduling based on priority the ATWT and ATTT are reduced when compared to Max-Min algorithm.

Software, Hardware Requirements and the Data used

The classified task scheduling method based on priority has been implemented using cloud simulation tool and Eclipse IDE. Different lengths of tasks different categories of virtual machines are assumed to measure the performance of classified task scheduling method. The performance of the proposed classified task scheduling method is measured against max-min scheduling algorithm. For the simulation setup we have considered/used the following.

Task lengths = {2000, 3000, 40000, 500000, 6000, 70000, 800000, 900000, 10000, 11000, 112000, 13000, 114000}.

All the virtual machines are assumed to be of similar configuration but

they can be different in real time. VM Description in Millions of Instructions per Second = {2500} Virtual Machines and Tasks are created randomly in order to compare with Max-Min Scheduling Algorithm. We have considered only 13 tasks for the simulation because of the limited resources available and if the resources are more then we can even create more number of tasks. Host RAM= 4GB, HOST HARDDISK=300GB, Host Bandwidth = 450Mbit/s and Number of CPU=1, Virtual Machine Monitor ="Xen" and Operating System = Linux.

Discussion on Parameters Considered

The parameters that are considered for evaluating the performance of the algorithm are Priority of the task, Total waiting time (TWT), Total Execution time (TET) and Total Turnaround time (TTT). From Table 2 and Table 3 we can see a great reduction in TTT when classified task scheduling based on priority is used for medium and high priority jobs. Because of one virtual machine handling high priority jobs we can see there is an increased TTT but average TTT is reduced which we can see from Table 4. How ever if we add more number of virtual machines for handling high priority tasks the average wait time as well as average turnaround time can be greatly reduced. However in the present scenario also we can see that Total wait time and Total turnaround time are reduced greatly. We can observe these comparisons from figure 3 and figure 4.

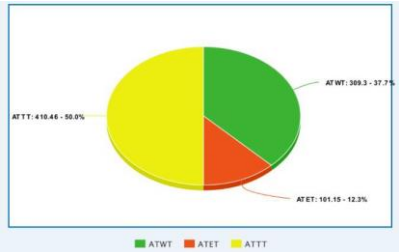

Figure 3. Task scheduling based on Max-Min algorithm.

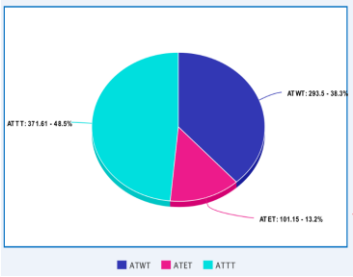

Figure 4. Classified Task scheduling based on priority.

From figure 3 and figure 4 we can say that average turnaround time of tasks is reduced.

Conclusion

In this paper we tried to attempt improve the max min task scheduling of IoT health care in fog environment by using classified task scheduling based on priority. In this attempt made by us we can see that average turnaround time of a job is reduced and starvation of low priority jobs are eliminated and long turnaround time of medium priority jobs are reduced. We can further improve the average turnaround time of a task by including more than one virtual machine in category 1. Further it is observed that number of virtual machines should be greater for handling high priority jobs. In this way it is observed that we can balance the load between virtual machines and all kind of patients will be able to get services.

References

[1] Shanzhi Chen, Huixu, Dake Liu, Bo Hu and Huchengwang, A vision of IoT: Applications, Challenges and Opportunities with China Perspective, IEEE IoT J., 1(4) Aug. (2014).

[2] Guangshun Li, Jiping Wang, Junhua Wu and Jianrong Song, Data Processing Delay Optimization in Mobile Edge Computing, Wireless Communications and Mobil Computing Journal, special issue, volume 2018.

[3] Gollaprolu Harish, S. Nagaraju, Basavoju Harish and Mazeeda Shaik, A review on fog computing and its applications, International Journal of Innovative Technology and Exploring Engineering, 8(6c2) April 2019.

[4] Lindong Liu, Deyu Qi, Naqin Zhou and YilinWu, A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment (2018), 1-12.

[5] Swati Agarwal, Shashank Yadav and Arun Kumar Yadav, An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing, I. J. Information Engineering

and Electronic Business 8(1) (2016), 48-61.

[6] Xuan-Qui Pham, Nguyen Doan Man, Nguyen Dao Tan Tri, Ngo Quang Thai and Eui-Nam Huh, A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing, International Journal of Distributed Sensor Networks 13(11) (2017), 1-16.

[7] L. Georgios, Stavrinides and Helen D. Karatza, A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments, Springer Science+Business Media, LLC, part of Springer Nature (2018), 1-17.

[8] Chaolin Huang, Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, articles, Lancet 395 (2020), 497-506.