Project Report on

Disaster Management Monitoring using Fog Computing in Internet Of Things (IoT) paradigm

Submitted by

Aditya Kumar	**B150476CS**
Atul Soni	**B150767CS**
Ramanuj Kumar	**B150887CS**
Stami Ray	**B150709CS**
Sruthi P. G.	**B150254CS**

Under the Guidance of

Rajasree T.

तमसो मा ज्योतिर्गमय

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

March 1, 2019

Disaster Management Monitoring using Fog Computing in Internet Of Things (IoT) paradigm

Aditya Kumar Atul Soni Ramanuj Kumar Stami Ray Sruthi P. G.

Abstract: Monitoring and prediction of disasters is one of the most imperative phases in disaster management. Critical, crowd-sourced, IoT data of a particular geographical region, when assessed and analyzed timely, enable effective preventive measures, prediction and early warning systems. In this project, we propose a disaster management monitoring system that analyses the crowd-sourced data using fog computing which avoids the latency and delay jitter that are usually experienced in a cloud computing platform making the latter unfeasible for time-critical applications. The architecture of the proposed project is inspired from the IEEE research paper, 'Crowdsourcing-based Disaster Management Monitoring using Fog Computing in Internet of Things paradigm' published by Rauniyar et. al. in 2016 [1].

1 Introduction

Internet of things (IoT) has gathered a lot of interest from both industry and research community. It is estimated by many experts that IoT will consist of around 50 billion objects by year 2020 and the amount of data generated by all objects will be very large. Cloud computing is widely used for the services such as analyzing crowdsourced data and applications implementation over the IoT. Using IoT and SaaS, crowdsourcing combines sensing technologies, analytical models, data management techniques and visualization methods to create solutions to improve the quality of human lives and its surrounding environment [1].
Nowadays, almost every country and humans are prone to natural and artificial disasters. IoT technologies can't stop disasters from happening, but can be very useful for disaster preparedness, such as prediction and early warning systems and rescue operation post disasters. In this way, IoT can compensate for a poor infrastructure.But if it were to be implemented in a cloud computing platform, the latency and jitter involved will render the application unfeasible and will make timely preparedness operations impossible. One of the possible ways to overcome this weakness of cloud computing paradigm is edge computing or fog computing.
One of the most popular compartmental models to evaluate the performance or effectiveness of disaster management monitoring systems is SIR(susceptible,infected,recovered) model. Like in any compartmental model, the population of the affected geographical region is divided into various compartments to analyse how they are being affected by the concerned event. In SIR model used in disaster management monitoring systems, the affected populace is partitioned into three compartments - namely, susceptible, infected and recovered. The gradient and the peaks of the three curves corresponding to S,I and R with time depict the related numbers.

2 Problem statement

In this project, we aim to simulate a crowdsourcing-based disaster management monitoring system using fog computing (CDMFC) model in IoT paradigm for early disaster detection and preparedness [1]. In the paper Crowdsourcing-based Disaster Management Using Fog Computing in Internet of Things Paradigm, proposed by Rauniyar et al, an architecture for implementing CDFMC is studied upon. Exploiting the advantages of crowdsourcing and fog computing to handle disaster management in an efficient way,the proposed CDMFC model can detect real-time disasters and disseminate early information for public safety, unlike the conventional cloud computing based systems. [1]. We would also plot SIR graphs using our CDMFC model in different scenarios of input data.

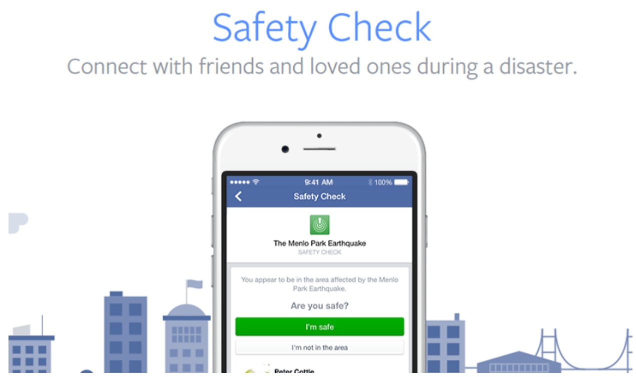

Figure 1: Safety Feature in Facebook

3 Literature Review

IoT, crowdsourcing and cloud computing provides a genuine platform to spread information early for the public safety during the natural or man made disasters. The authorities responsible for public safety can launch rehabilitation operation for disaster affected people and areas.

3.1 Safety Check and Community Help in Facebook

During the major earthquake in Nepal, on 25 April 2015, killed many people and many were injured . Soon, many people updated the live events of earthquake through messages, pictures or videos in social media. The crowdsourced data through social media were useful to launch rehabilitation program to reach out to the needy people. People also need to know about the safety of loved ones.
That time Facebook invented a feature called Safety Check. This feature is based on the crowdsourcing model to spread the information as soon as possible so that effective rescue can be planned accordingly. [6]

Working principle of 'Safety Check' feature :
(i) When Safety Check feature is activated, it finds users near a disaster site from where they used the internet or through the place listed on their profile.
(ii) Facebook users are then asked to confirm whether they are safe or not.
(iii) Those Facebook users who choose option safe, create a notification message to their family members, and friends.
(iv) Users who choose safe option can also track how many of their friends or family members were affected. [1]
In 2017, Facebook started a Community Help feature to the Safety Check crisis response tool. It allows users to search through categorized posts, offer local assistance, and connect with providers over Facebook Messenger. After that Facebook announced one update in Safety Check to start raising funds. [6]

3.2 5W Model

The 5W models are usually mentioned in journalism, research and police investigations. According to the principle of the Five W, a report can only be considered complete if it answers these questions starting with an interrogative word. [7]
5 Questions are as follows:
(i) Who was involved?
(ii) What happened?
(iii) Where did it take place?
(iv) When did it take place?
(v) Why did that happen The 5W model has been suggested for managing urban emergency events. The target of crowdsourcing are the users of social media. But, this is based on cloud computing platform which has the disadvantage of high latency. An efficient method to deliver time-critical sensing data successfully to the cloud during disasters with poor

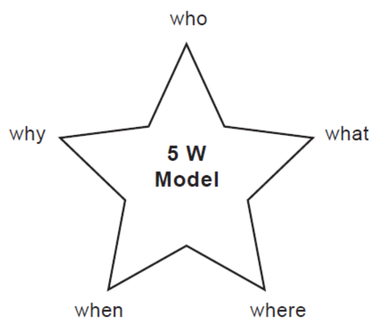

Figure 2: 5W Model

communication architecture has not been researched so far. [1]
In disaster management monitoring field, 5W model is used to analyse the social media big data. The post,updates,any multimedia data like photos, videos are analyses on the basis of 5W model. This can be useful to detect any disaster or emergency-related events happening in a particular geographic region. The importance of social media as an information hub is overwhelming. 5W model enables employing this vast amount of data to extract beneficial information for timely response and preparedness activities. In this way, most of the urban population can be notified of the event updates in spite of the breakdown of the conventional communication grid.

3.3 Micro-Message Delivery Communication Service

Immense deployment of IoT devices can bring benefits in data network resilience during the disaster period. The conventional communication service can be out of service during a disaster. The IoT devices can enable emergency micro-message delivery communication service through data prioritization schemes. [1]
Sensing data by the mobile users were crowdsourced to the cloud platform for analysis of public safety. The mobile devices for sensing were chosen through auction. Money is also given to the sensing data providers. The researchers utilized the IoT data and contextual information to enhance and manage the emergency situations.

4 Architecture/Framework

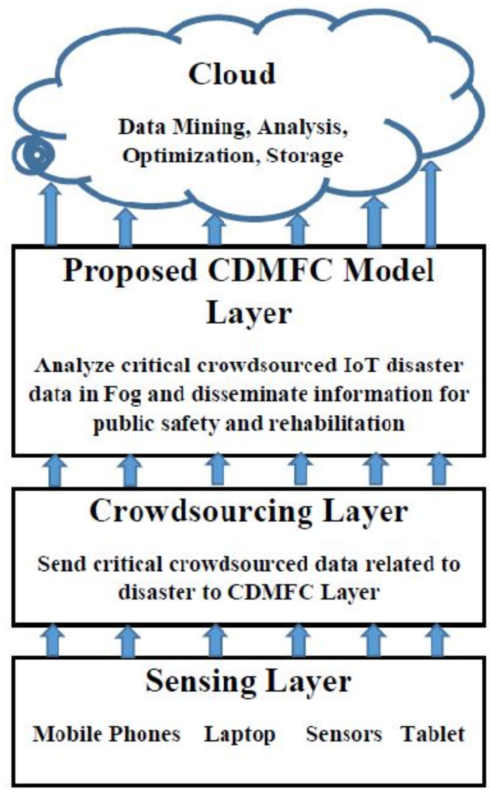

Figure 3: Hierarchical Layered architecture

CDMFC model views the whole system as a hierarchical structure, in which logical modules or subsystems at different layers in the hierarchy work autonomously for the proper functioning of the disaster management monitoring system. CDMFC model is decomposed into four layers namely sensing, crowdsourcing, CDMFC and cloud computing layer. In this hierarchical design, a low-level subsystem gives services to its adjacent upper level subsystems, which invoke the methods in the lower level.
The dependency between the modules is based on the fact that they communicate by passing only data. The components are independent to each other and communicating through data. Therefore the modules are data coupled. When practically implemented, these modules interact with each other using designated APIs or interfaces which are used for abstrac-

tion of datatype conversion, size truncation, etc.

5 Methodology/Working of the system

CDMFC model follows a hierarchical, layered architecture comprising four layers namely :- (i) Sensing layer, (ii) Crowdsourcing layer, (iii) CDMFC layer and (iv) Cloud computing layer. Each layer has autonomous functions to do, and works together to facilitate the working of the system. The detailed working of each layer is explained below :-

5.1 Sensing Layer

This layer is the first layer of CDMFC layered model. This layer senses different events such as fire, earthquake, flood, other natural and artificial disasters. This is possible via a multitude of IoT applications through sensors, mobile phones,laptops, tablets etc. This paper also mentions the term - social sensors - which is actually the event-related data generated by humans through mobile phones and tablets. This layer has to deal with the largest amount of data among all the layers. This is because the data generated in this layer may not all be related to disasters or emergency events. The primary duty of the sensing layer is only to sense data, irrespective of the type of event the data is related to, and pass it on to the next layer in the hierarchy, crowdsourcing layer.

5.2 Crowdsourcing Layer

The second layer of the CDMFC model, crowdsourcing layer, deals with the crowdsourcing or collecting all the sensing data generated from the sensing layer.In Internet Of Things paradigm, crowdsourcing is the process of obtaining and analyzing information or input to a particular task or project generated by a number of sources such as sensors, mobile devices, vehicles and humans.Conventionally in most of the conventional works, it is very common to send the crowdsourced data to the cloud directly, where further analysis using data mining and visualization techniques take place. This is done to make a knowledge base for arriving at a final conclusion about related events.

Since the amount of crowdsourced data from billions of IoT devices will be extortionate, the processing in cloud computing platform is time-consuming and suffers from latency and delay jitter.This can adversely affect the purpose of the project which was timely prediction and aversions of damage to lives and property in the events of disasters and emergency events. Therefore, in the paper proposed by Rauniyar et. al [1], the CDMFC model sends the data related to disasters and emergency events directly to the CDMFC layer for effective disaster management monitoring for public safety.

To achieve this goal of sending only the disaster-related data to CDMFC layer, CDMFC model adopts filtering technique based on emergency and disaster-related keywords generated by the IoT applications and humans through different mobile phones and tablets ands ensors deployed in disaster-prone region or location. Thus, this layer refines the data that are pulled in from the sensing layer.

5.3 CDMFC Model Layer

It is in this layer that the crowdsourced critical disaster-related IoT data is analyzed. Exploiting the advantages of fog computing,the analysis is performed on the data in a distributed way in real time. In conventional systems where huge amounts of data are sent directly to cloud computing layer from the crowdsourced layer which eventually leads to unavoidable latency issues, the proposed CDMFC layer redeems the system from such delays which facilitates early warning and timely preventive actions in case of disasters or emergency situations. The data is always associated with the corresponding time and locations stamps which enable the system to exactly identify the origin and time of any noticeable event.

The CDMFC layer is directly accessible to public safety authority and contains emergency contact numbers, which aids in planning out necessary action according to the crowdsourced data received from a particular geographical region. The fog maintains all the disaster-related data in whichever form it is

available, be it video, clips, photos, phone numbers, so that people can easily access and be informed of developments in their area. This storage of disaster-related data in the fog, close to where it is generated also helps to conserve bandwidth.
In any application involving IoT, security is a major concern. Since most of the disaster-related data is to be stored in the fog, deployment of light-weight security algorithms will help to curb the security menace.

5.4 Cloud Computing Layer

Though CDMFC layer plays the most important role in our project for the purpose of disaster management monitoring, the benefits of cloud computing layer cannot be ignored. The non-critical data from crowdsourcing layer and other data from CDMFC model layer are sent to the cloud computing layer. The data is analyzed and stored in the cloud computing layer for processing at a later instant of time. In this layer, extensive data mining and visualization techniques are applied to arrive at concrete and final conclusions about the events concerned. Further, cloud computing layer enables data to be stored for a long time in the cloud for historical analysis.

6 Dataflow diagram

Each layer is simulated as a process. Therefore the four layers in the hierarchical architecture that have been proposed will be represented as four processes namely- sensing process, crowdsourcing process, CDMFC process and cloud computing process. The sensing data enters the second process- crowdsourcing process from the sensing process. After the filtering process employed in crowdsourcing process, only the critical disaster-related data is being sent to the third process, CDMFC process. The CDMFC process will have subroutines which communicate with the public safety autoritied or modules that are dedicated to notify the public about any imminent disasters. The non-critical data from the CDMFC layer and also the crowdsourcing layer is passed on to the fourth process- cloud computing -process where the data may be stored on a long-term basis for historical analysis.

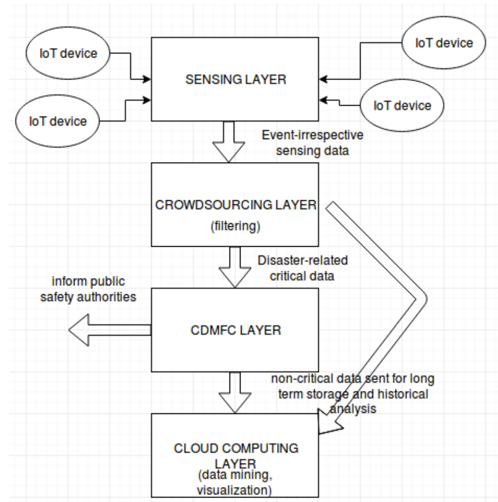

Figure 4: Dataflow diagram representing the flow of data between various modules of CFMFC Model

7 Algorithms proposed

The functions that are executed in the CDMFC model can be summarised as below:
1.**Event Sensing** : Sensing layer (Layer I)
2.**Crowdsourcing data** : Crowd sourcing layer (Layer 2)
3.**Sending Disaster related IoT Data to Fog/CDMFC Layer** : Crowd sourcing layer (Layer 2)
4.**Analyzing Disaster related IoT data in Fog** : CDMFC layer (Layer 3)
5.**Disseminating Early Information for Public Safety** : CDMFC layer (Layer 3)
In the crowd-sourcing layer, the filtering of the IoT data based on keywords related to disasters is done. We will use a modified version of Knuth-Morris-Pratt(KMP) algorithm for matching the keywords and extracting relevant data.
Algorithm for modified KMP used in keyword-based filtering in crowdsourcing layer:-
Here the text refers to the entire data received by crowdsourcing layer from the sensing layer. Patterns refer to the keywords related to the disasters. If a

match is found, i.e., if disaster related keywords are found in a particular quantum of data, then that data is sent to the CDMFC for futher analysis, otherwise it is either discarded or sent to the crowdsourcing layer.
Step 1: Initialize the input variables :
n = Length of the Text
m = Length of the Pattern
u= Prefix function of pattern (p)
q = Number of characters matched
Step 2:Define the variable :q=0, the beginning of the match
Step 3: Compare the first character of the pattern with first character of text. If match is not found, substitute the value of u[q] to q. If match is found, then increment the value of q by 1.
Step 4: Check whether all the pattern elements are matched with the text elements. If not, repeat the search process.If yes, send the related data to the CDMFC layer.
Step 5: Look for the next match.
The running time of the KMP algorithm is optimal (O(m+n)), which is very fast. Here m stands for the length of the keyword and n stands for the length of the text.The algorithm never needs to move backwards in the input data. It makes the algorithm good for processing very large files.

8 Other Implementation Details

The architecture proposed in the research paper,'Crowdsourcing based disaster management using fog computing in Internet of Things (IoT), by Rauniyar et. al. is layered and hierarchical. The language of implementation will be Python 2.7. Each layer is represented as a process which communicates to another process that mimics the immediate upper layer in the hierarchy. The four layers in the hierarchy will have corresponding four processes that will be implemented in python.
Dataset is speculated to be taken from ASEAN Disaster Information Network (ADInet). We shall also endeavour to take real-time data using IoT enabled smart phones during later staged of implementations.
SIR graphs are plotted against each run of the system on different input datasets. SIR is one of the most popular compartmental model used to analyse the different disaster management monitoring models based on their effectiveness. Figure 5 shows an example of how SIR plot looks like.

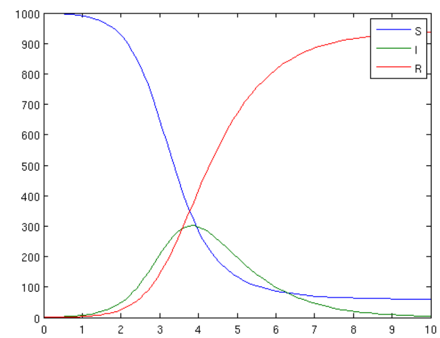

Figure 5: SIR Plot showing the respective susceptible, infected and recovered figures

9 Conclusion

Crowdsourcing the IoT data related to disasters can help public safety authorities to plan out efficient rescue operation and saves millions of peoples lives stuck in such dreadful situations.However, crowdsourcing data are often analyzed in cloud platform where latency will be quite high.Therefore, we need to minimize latency for better public safety and management during a natural disaster and emergency situations.The proposed CDMFC model for disaster management which takes the advantage of fog computing platform where the time-critical crowd-sourced IoT data related to disasters is analyzed in real-time. Through our CDMFC model, we can detect the disasters in real-time and plan out response operations accordingly.CDMFC model can securely operate on the IoT data inside the fog where we can install of our own lightweight security algorithms. Therefore, security feature of the system can be enhanced.

References

[1] Rauniyar, Ashish Engelstad, Paal E. Feng, Boning Thanh, Do. (2016). Crowdsourcing-based Disaster Management Using Fog Computing in Internet of Things Paradigm. 490-494. 10.1109/CIC.2016.074, https://ieeexplore.ieee.org/document/7809745

[2] Lambrinos, Lambros. "On combining the Internet of Things with crowdsourcing in managing emergency situations." Communications (ICC), 2015 IEEE International Conference on . IEEE, 2015, https://ieeexplore.ieee.org/document/7248387

[3] Dubey, Rameshwar, et al. "Developing an integration framework for crowdsourcing and Internet of Things with applications for disaster response." Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference, https://ieeexplore.ieee.org/document/7396552

[4] Han, Shuihua, et al. "Harnessing the power of crowdsourcing and Internet of Things in disaster response." Annals of Operations Research (2018): 1-16, https://link.springer.com/article/10.1007/s10479-018-2884-1

[5] Sinha, Akash, et al. "Impact of internet of things (IoT) in disaster management: a task-technology fit perspective." Annals of Operations Research (2017): 1-36, https://link.springer.com/article/10.1007/s10479-017-2658-1

[6] "Facebook Safety Check", https://en.wikipedia.org/wiki/Facebook-Safety-Check (accessed Feb. 28, 2019).

[7] "Five Ws Model" https://en.wikipedia.org/wiki/Five-Ws (accessed Feb. 28, 2019).

[8] Bhagya Sri, M., Bhavsar, R., Narooka, P. (2018). String Matching Algorithms. International Journal of Engineering and Computer Science, 7(03), 23769-23772. Retrieved from http://www.ijecs.in/index.php/ijecs/article/view/3995

[9] Duan, Geng L et al. The implementation of KMP algorithm based on MPI + OpenMP. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (2012): 2511-2514.

[10] Li, Chao, et al. "Edge-oriented computing paradigms: A survey on architecture design and system management." ACM Computing Surveys (CSUR) 51.2 (2018): 39.