International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor (2018): 7.426

Cloud-Based Disaster Management Using Mobile Edge Computing Mechanism

Arunavo Dey[1], Nahid Anwar[2]

Abstract: ***Social sensing services use humans as sensor carriers, sensor operators and sensors themselves in order to provide situation-awareness to applications. This promises to provide a multitude of benefits to the users, for example in the management of natural disasters or in community empowerment. However, current social sensing services depend on Internet connectivity since the services are deployed on central Cloud platforms. In many circumstances, Internet connectivity is constrained, for instance when a natural disaster causes Internet outages or when people do not have Internet access due to economical reasons. In this paper, we propose the emerging Fog Computing infrastructure to become a key-enabler of social sensing services in situations of constrained Internet connectivity. To this end, we develop a generic architecture and API of Fog-enabled social sensing services. We exemplify the usage of the proposed social sensing architecture on a number of concrete use cases from two different scenarios***

Keywords: Social Sensing, Edge Computing, Situation Awareness

1. Introduction

Situation-aware applications use data streams from sensors to provide useful services to users or other applications. With the proliferation of sensors deployed in the surrounding world, e.g., through the Internet of Things, the potential of such applications is reaching new dimensions. Recently, research focus has been expanded from traditional fixed sensor deployments toward social sensing [1]. This comprises passive sensors provided by human carriers in Smart Phones, active human sensor operators taking pictures or videos and even humans operating as sensors themselves, e.g., providing live information in tweets and postings. Recently, new applications have been proposed which use the social sensing infrastructure to infer situations that are not detectable from traditional sensors.

1.1 Application Fields

An important application field of social sensing is in helping people to deal with natural disasters. There are applications that help in finding friends and family in the aftermath of a natural disaster[11].Furthermore, social media can provide access to relevant and timely information to individuals in affected regions [17]. Providing real-time information to disaster-affected people about the situation in the area can help them take mitigative actions, for instance moving contents located in a flood-prone ground floor to upper floor [2] to reduce the loss caused by the disaster. Social media has been an effective way of sharing this sort of crowd-sourced information and can be more accurate and meaningful than government predictions. Many proposals envision disaster-stricken people to perform social sensing tasks, like providing information about the level of inundation of roads in the event of a flood or tsunami. Such un-structured information would be mined by a social sensing application to extract relevant details and create a map of the affected area with important information. These maps can be used by government agencies to perform rescue operations [8]. Users can upload pictures of people with them, and social sensing applications apply face recognition algorithms on the pictures and let the friends and family of detected individuals know that they are safe. In rural or economically under-served regions, social sensing helps in understanding socioeconomic processes [11] which can empower communities to better utilize their social capital1 and enable self-organized governance. Public transportation in such regions leave much to be desired due to lack of

Figure **??** consistency in schedules and infrastrual sup-port, forcing passengers to wait for long periods of time. In well-served communities, infrastructual support (e.g., kiosks at bus stops operating on GPS data) provide timely information for the passengers. Social sensing services in such under-served regions could help gather information, e.g., when the bus is going to arrive and share with others even in the absence of infrastructural support.

1.2 Challenges

While the discussed applications are very effective in utilizing social sensing information, they rely on Internet connectivity of the social sensors, the situation inference applications, and the users that are interested in the detected situations. is is mainly the case because the social sensing service is hosted in a central (cloud) data center.

However, Internet connectivity cannot be taken for granted on any of the layers of a social sensing application. Internet outages can affect large areas in case of emergencies, natural disasters, or hacker attacks on the Internet infrastructure [13]. Furthermore, rural regions might not be connected to the Internet at all, or the inhabitants of a rural or an under-served urban region cannot afford Internet connectivity for economical reasons. Social sensing applications can be of a huge benefit in exactly such situations and circumstances. All of those benefits are tightly coupled to the Internet connectivity; without the Internet, social sensing services are not available. In recent years, a new trend has emerged in computing infrastructures that can help in overcoming the Internet dependency of social sensing services. Fog Computing, also known as Edge Computing, is the approach of adding computational resources toward the edge of the Internet

[5]. While it was initially intended to improve network latency between sensors, applications, and users [10], we propose Fog Computing to become a central enabler of

decentralized, local social sensing services that can also operate when Internet connectivity is constrained. This way, social sensing services can become more robust to Internet outages. Furthermore, communities that did not benefit from the first wave of cloud-based social sensing services can leapfrog those and directly use Fog-based services.

However, today social sensing services are not capable of using the Fog infrastructure to provide local services when Internet connectivity is impaired. It is not enough to just run a centralized social sensing service on a number of Fog nodes in parallel. Instead, the social sensing service has to become a distributed service capable of discovering available Fog nodes and building a network that aggregates and shares information between social sensors that are connected to different Fog nodes. In this regard, it needs to be able to deal with the volatile nature of Fog and sensor connectivity. To this end, the architecture of social sensing services needs to be adapted to fully utilize the opportunities of the Fog infrastructure.

1.3 Outline

In this paper, we give an overview of evolving Fog-based computing infrastructures. Based on that, we propose a generic architecture for Fog-based social sensing services. Using two concrete case studies, we demonstrate how existing cloud-based social sensing services can be adapted to use the Fog-based architecture. We conclude that utilizing Fog-based computing architectures is a promising path to more robustness and democratization of social sensing services.

2. FOG-Based Warning Architecture

In the following, we give an overview of the emerging Fog Computing architecture. We point out that the Fog infrastructure can be completely heterogeneous. Social sensing on Fog has to be able to cope with the heterogeneity provided in the available resources.

Figure 1 shows a model of the Fog Computing architecture. On the top layer, the traditional Cloud data center is depicted, being deployed in the core of the network and only reachable via Internet connections. Such data centers are characterized by using standardized, o-the-shelf computing resources, and a virtualization layer that allows for an effective utilization of the resources and a pay-as-you-go business model. In the middle layer, a number of heterogeneous Fog nodes are geographically distributed deployed at the edge of the network. is means, that Fog nodes can be locally reachable by connected devices nearby, even if the Internet is not available. On the boŁom layer, geographically distributed social sensors are connected to their close-by Fog nodes, either directly or by using other social sensors as relays.

As there is a varied uses of Fog computing, there are many different notions of a Fog node. In the following, we provide an overview of current proposals and products

With the advent of computationally stronger network equipment, especially routers, it has been proposed that computations are already performed in the network. For instance, Cisco offers their IOx platforms on hardened routers [6] that are capable of performing data processing tasks. On a higher layer, mini-computers like Raspberry Pi have gained popularity, as they provide acceptable computation performance for a very low price. Additionally, the energy efficiency and miniaturization of those devices allow them to run in environments that were not specifically designed to host computers, i.e., outside of data centers. Mini-computers can even be deployed on drones [9] and provide a completely new level of mobile computing. A swarm of drones can build an ad-hoc network, a so-called Flying Ad-Hoc Network (FANET) [21], and this way provide Fog computing in an area that lacks any infrastructure. Generally, the deployment of Fog services can be facilitated by using recent lightweight container technology like Docker [4]. e social sensors can be smart sensors that perform the sensing, but also altering and aggregation. In the scenarios described, typically the smart sensors would be connected to smart phones which have certain computational capabilities to do the altering and aggregation. This reduces the communication overhead between social sensors and Fog nodes, and also reduces computational overhead on the Fog nodes.

2.1 Fog-enabled social warning services

Here, we analyze how social warning services can exploit the Fog infrastructure. They should be able to operate on local information provided on a single Fog node, but also capable of sharing information and collaborating with social sensing services running on neighboring Fog nodes that are reachable. Finally, if the Cloud is reachable, the social warning services on the different Fog nodes should be able to share global information via the Cloud.

We propose a generic software architecture for social warning applications that is capable of exploiting the Fog infrastructure (cf. Figure 3). It consists of three components:

(i) A central management components placed in the Cloud infrastructure (the Cloud Component), (ii) A data process-ing component placed in the Fog infrastructure (the Fog Component) and (iii) a social sensing component deployed on the users devices (the Sensor Component). In the following, we detail the tasks of the components .

2.1.1 Cloud component

The Cloud Component is mostly responsible for the deployment and management of the Social Warning Service artifacts (program code, meta-data, settings, etc.) on the cloud. The cloud worked here as a consistent database storing all the information needed and updated. Fog Components, when an Internet connection is available. data is loaded from the cloud and fog components are updated so that for future warning services they can predict on their own.

2.1.2 Fog component

The fog component worked here for verifying the flash warning generated by any sensor component and circulat-ing the warning to its nearby sensor.A generated warning from any sensor, when send to a fog component, needs to be verified with the existing information. Depending on the type of the disaster, types of information needed for verifi-cation

are different.The cloud made sure that the available information is present and whenever a fog node updated, it updated its latest information needed to verify the warning and during disaster time, if cloud goes offline, any message from sensor,coming to any fog node, can be verified with the help of previous information and circulated.

2.1.3 Sensor component

The Sensor Component are here the same devices who work as fog nodes with the difference that they don't need the same calculation powers as fog nodes have and only work here is to generate flash warnings to be sent to fog components. It should be noted that not all Sensor Components might be able to directly connect to a Fog Component. The defect for connection could have been their physical distance to the next Fog Component, or device limitations (e.g., supporting the communication requirements of the Fog Components). For instance, if the Fog Components all require 4G connectivity, some of the Sensor Components might not be able to directly connect to any Fog Components at all. But in this project, all Sensor Components could connect to other fog Components in their proximity, for instance, using WiFi networking or @G connections. Such a network can, for instance, be realized with methods from Mobile Ad-Hoc Networks (MANETs). A similar idea was presented by Yusuf, et al [20] with the micrograph middleware. It shows how to handle discovery and manage these distributed and isolated communities for social networks. Note that as the Sensor Components can be disconnected from the Fog at any time, e.g., because the Fog Component goes down, continuous queries on the Sensor Components should be so state, i.e., employ a time-out mechanism; when the connection to the Fog layer is interrupted for a long time, the sensing is stopped to save energy on the social sensing devices.

3. Case Studies

Social warning services deployed on the Fog can help to gather and disseminate local knowledge among the affected people. Owing to the relatively local nature of the information pertaining to a disaster-prone area, Fog Computing is destined for providing the required connectivity to affected people and so that they can help mitigate the adverse effects.

As it is guided by the information it receives, to narrow down the disaster category for warning is necessary for its operational purpose. We operated our proposed system for flood warning service.

3.1 Warning for flood

In a flood prone area, any sensor who can get any data regarding sudden rise about the water level can send an alarm message to the nearest fog component. A fog component, whenever connected to the internet, gets updated with the latest list of devices it serves. Whenever an alarm is received by this fog component, it creates an alarm in the fog device. Upon receiving the alarm, if the owner of the fog device decides whether the verify process should be turned or not. If verify process is turned on, then the algorithm for verification runs and yields its result. If the result is above then a threshold value, the all the recipients from that list of fog component receives a warning message.

Algorithm 1 Verification algorithm

```
1:  procedure VERIFY
2:  level←levelofwater
3:  req←Requests
4:  i←patlen
5:  top:
6:  if level==danger then x=1.0.
7:  if level==strong then x=0.5.
8:  if level==normal then x=0.0.
9:  if req >1 then y=1.0.
10: if req== 1 then y=0.5.
11: if req <1 then y=0.0.
12: if (0.7x +.3y) ¿threshold then verify.
```

4. Technical Challenges

The Fog infrastructure poses a large range of technical challenges on the implementation. For example, if the Fog nodes are installed on drones, different communication protocols are used and coupling between them is required. Additionally, the network protocols need to be latency-tolerant; each node needs to be able to queue messages until a connection is reestablished. Handling geo-distributed resources is challenging. Part of the complexity is defining the type of algorithm to deploy on the nodes based on the available capabilities. is has to be added to the process of deploying applications to nodes with limited Internet connectivity and untrusted infrastructure. Common distributed systems issues also arise in the context of Fog social sensing. Fog resources might have lower availability and dependability than servers in cloud data-centers. One of the main challenges is that protocols and middleware need to be distributed and energy efficient, e.g., discovering other peers and fog nodes without a central entity and with limited energy. Load balancing is another common issue. For example, the region of a disaster may require more resources such as networking and computing. How can the Fog infrastructure be organized to meet different resource demands? Mobility of Fog nodes could be used to dynamically balance the pressure on each Fog node. There also exist social sensing specific challenges. Location updates ofa sensor node may lead to errors necessarily unintended, but caused by mistakes.

Fog computing itself enhances the sharing of information within the region responsible for a given Fog node. However the question arises, is there more we can do to provide reliable information sharing? For example, an intuitive idea is to gather the information from different social sensors and eliminate outliers. A further question is how to route the information to the fog component

5. Conclusion

In this paper, we have extended the vision of Fog Computing toward providing social sensing services in situations when Internet connectivity is limited. We have outlined the basic design principles of such a Fog-enabled social sensing service, and have proposed a generic solution that social

warning services can employ in order to use the Fog infrastructure.

References

[1] H. Kopka and P. W. Daly, A Guide to LATEX, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] Charu C Aggarwal and Tarek Abdelzaher. 2013. Social sensing. In Managing and mining sensor data. Springer, 237297.
[3] Maura C Allaire. 2016. Disaster loss and social media: Can online information increase ood resilience? Water Resources Research 52, 9 (2016), 74087423.
[4] Arvind Arasu, Shivnath Babu, and Jennifer Widom. 2006. e CQL Continuous ery Language: Semantic Foundations and ery Execu-tion. Łe VLDB Journal 15, 2 (June 2006), 121142.
[5] Paolo Bellavista and Alessandro Zanni. 2017. Feasibility of Fog Computing Deployment Based on Docker Containerization over RaspberryPi. In Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN 17). ACM, Article 16, 10 pages.
[6] L. Yusuf and U. Ramachandran. 2012. Community Membership Management for Transient Social Networks. In 2012 21st Interna-tional Conference on Computer Communications and Networks (ICCCN). 17.
[7] lker Bekmezci, Ozgur Koray Sahingoz, and amil Temel. 2013. Flying Ad-Hoc Networks (FANETs): A survey. Ad Hoc Networks 11, 3 (2013), 1254 1270.
[8] R. Stiegler, S. Tilley, and T. Parveen. 2011. Finding family and friends in the aermath of a disaster using federated queries on social networks and websites. In 2011 13th IEEE Intl Symposium on Web Systems Evolution (WSE). 2126.
[9] Magnus Skjegstad, Frank T Johnsen, Trude H Bloebaum, and Torleiv Maseng. 2012. Mist: A reliable and delay-tolerant pub-lish/subscribe solution for dynamic networks. In New Technolo-gies, Mobility and Security (NTMS), 2012 5th International Con-ference on. IEEE, 18.