LivingFog: Leveraging fog computing and LoRaWAN technologies for smart marina management (experience paper)

Davaadorj Battulga, Mozhdeh Farhadi, Mulugeta Ayalew Tamiru, Li Wu, Guillaume Pierre

LivingFog: Leveraging fog computing and LoRaWAN technologies for smart marina management (experience paper)

Davaadorj Battulga
U-Hopper, Univ Rennes, Inria, CNRS, IRISA
davaadorj.battulga@u-hopper.com

Mozhdeh Farhadi
U-Hopper, Univ Rennes, Inria, CNRS, IRISA
mozhdeh.farhadi@u-hopper.com

Mulugeta Ayalew Tamiru
Elastisys AB, Univ Rennes, Inria, CNRS, IRISA
mulugeta.tamiru@elastisys.com

Li Wu
Elastisys AB, TU Berlin
li.wu@elastisys.com

Guillaume Pierre
Univ Rennes, Inria, CNRS, IRISA
guillaume.pierre@irisa.fr

***Abstract*—In recent years, fog computing has emerged as a paradigm that brings computing, storage and networking resources closer to end users and devices at the edge of the network. One of the use cases for fog computing is IoT, where a large amount of data is generated by sensors that need to be pre-processed in place before the results are sent to the cloud for further processing and long-term storage. However, actual fog deployments are at their infancy. In this paper, we present the smart-marina project at La Marina de Valencia in which the LivingFog fog computing platform integrating open-source software and LoRaWAN technologies were used to process data collected from several sensors. We show the benefits of the platform in terms of latency reduction and bandwidth saving. Moreover, the platform has been used by particpants of the "Hack the fog" hackathon to deploy applications to test different innovative ideas on using the sensor data.**

***Index Terms*—Fog computing, LoRaWAN, Kubernetes**

I. Introduction

The Internet of Things (IoT) is transforming several industries into smarter ones. In such situations, data collected from a number of sensors is used to generate insights that can be used to improve business decisions. Among the industries that are using IoT technologies are marinas, which traditionally were focused on sports activities, but are increasingly becoming living spaces and host cultural activities.

Traditionally, IoT projects are based on proprietary, application-specific or siloed, vertical solutions that require independent management ecosystems with poor integrations across domains. Moreover, many IoT solutions rely on remote cloud data centers for data storage and processing. This approach suffers from high latency, low bandwidth and high cost of data transportation.

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 765452. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

In the last few years, fog computing has been proposed to reduce the latency and cost of delivering data to a remote cloud by bringing computation close to the sources of data [1]. As such, fog computing complements IoT deployments by co-locating computation with a group of sensors and allowing pre-processing of data in real-time, thus reducing latency and the amount of data sent to remote clouds.

Several fog computing platforms based on open-source technologies have been proposed. Among them, the LivingFog platform that integrates LoRaWAN technology with Kubernetes has been used for potable water management in València, Spain [2]. This open-source platform has the potential to support sensors from multiple vendors, thus avoiding vertical siloes and independent management. Therefore, in this work, we extend the LivingFog platform to support multiple sensors in a smart marina setting at La Marina de València.

La Marina de València is a marina that is found in the Eastern part of Spain facing the Mediterranean Sea. The marina has two docks with mooring births, several restaurants, and public spaces. It is famous for having hosted the 32nd and 33rd America's Cups in 2007 and 2010.

In the past, the marina deployed few sensors: a meteorological station to monitor weather conditions, security cameras to monitor the area and stream to online visitors, people counters to measure the number of people coming to the area, as well as vehicle counters to measure the number of vehicles.

The FogGuru project, in collaboration with Las Naves, worked in this project to realize the marina's desire to expand the existing sensors and deploy additional ones. In a project that lasted seven months starting from September 2020, several sensors have been deployed at La Marina de València to measure parameters such as sea wave and current, water quality, weather conditions, number of people and number of vehicles. We also deployed a fog computing platform that enabled the processing of the sensor data in real time.

This project has four objectives:

1) Extend the LivingFog platform to support an integrated management of multiple sensors types [2].
2) Help La Marina de València realize its desire to setup an IoT and fog computing platform.
3) Address research and technical challenges such as resource management, multi-tenancy, security and data processing.
4) Enable a hackathon and IoT fablab so that citizens can exploit the data generated by the sensors themselves.

The rest of the paper is structured as follows. In Section II, we review some of the works in the literature that are similar to ours. In Section III, we discuss the requirement analysis for the project, the design decisions and the details of the implementation. In Section IV we use experimental measurements to show the advantages of the LivingFog platform in terms of reducing latency and reducing the number of messages that are sent to the cloud. In Section V, we describe some of the use cases based on our solution that were proposed by participants of the Hack the Fog hackathon. Finally, we conclude the paper in Section VI.

II. Related Work

In recent years, the Internet of Things (IoT) has drawn significant attention from research and industry [3]. IoT integrates several technologies and communication solutions [4]. Using these technologies, different objects such as RFID tags, mobile devices, sensors and actuators interact and cooperate to achieve common business goals with minimal human intervention [4], [5]. IoT technologies are enabling several industries such as smart cities, smart homes, e-health, smart grid, manufacturing, security and smart grid.

One of the defining characteristics of IoT is the large amount of data generated by a vast number of sensors and mobile devices. These data need to be stored and processed to get insights and visualizations, and to be able to make decisions based on them. As IoT devices are mostly battery operated and have limited data storage and processing capacity, they need to be complemented by more powerful resources. Moreover, these devices cannot host application services directly due to the said resource constraints [6]. As a result, traditionally, the data generated by IoT devices is usually transported over long distances to centralized cloud data centers for processing and long-term storage.

Although centralized cloud computing data centers provide vast amounts of processing and storage capacity, they are concentrated in few locations that are considerably far away from the vast majority of IoT sensors, actuators and mobile devices. Such non-negligible distances from IoT devices results in some drawbacks such as high latency, congestion of narrow-bandwidth networks and high cost of data transport over the network. To address these drawbacks, the fog computing paradigm is proposed, which extends the cloud by distributing computing, networking and storage resources to the network edge where data is generated [7], [8].

Despite fog computing being discussed vastly in academia, there are only few real-world implementations that are reported. *Byers* presents several requirements that a fog computing platform needs to fulfill and matches these requirements to several use cases from different industries. The author also presents a high-level software architecture for fog platforms without discussing implementation details and technologies used to achieve the platform [9].

Noghabi et al. explore several enterprises that use fog/edge computing presently in different industries such as business, smart cities, transportation and industrial plants [10]. The authors explain the motivation behind the used of edge/fog computing in these enterprises and present a typical edge/-fog deployment architecture without going into details about implementation and technologies used to achieve the specific deployments.

Yannuzzi et al. present a converged cloud/fog paradigm to address physical, data, service management and administrative siloes that smart city solutions suffer from, which they implemented in the city of Barcelona, Spain [11]. The authors present the architecture of the solution based on the *ETSI MANO* orchestration stack and five use cases where it was used.

Arkian et al. present a fog computing architecture based on open-source software integrated with LoRaWAN networking for potable water management in the city of València, Spain [2]. However, the authors do not go into the details of the implementation of the platform apart from identifying the challenges that the platform addresses. Our work is based on this platform and extends it to support multiple sensors, multiple fog computing clusters, multi-tenancy and data management in the context of smart-marina management at La Marina de València.

III. Design and Implementation

In this section, we discuss in detail the design and implementation of the IoT and fog computing platform at La Marina de València.

A. Requirements analysis

The first requirement is to select sensors that capture the following parameters:

- Sea wave, current and related parameters
- Wind speed and direction
- Sea water quality
- Outdoor weather conditions such as temperature, humidity, air pressure, rainfall level,
- Indoor environment conditions such as ambient temperature, humidity, O2 level, CO2 level and air quality
- Number of vehicles
- Number of pedestrians

The second requirement includes providing a fog computing platform for processing the data received from the several sensors as they arrive and make them readily available in a human readable format for the management of La Marina and other interested parties.

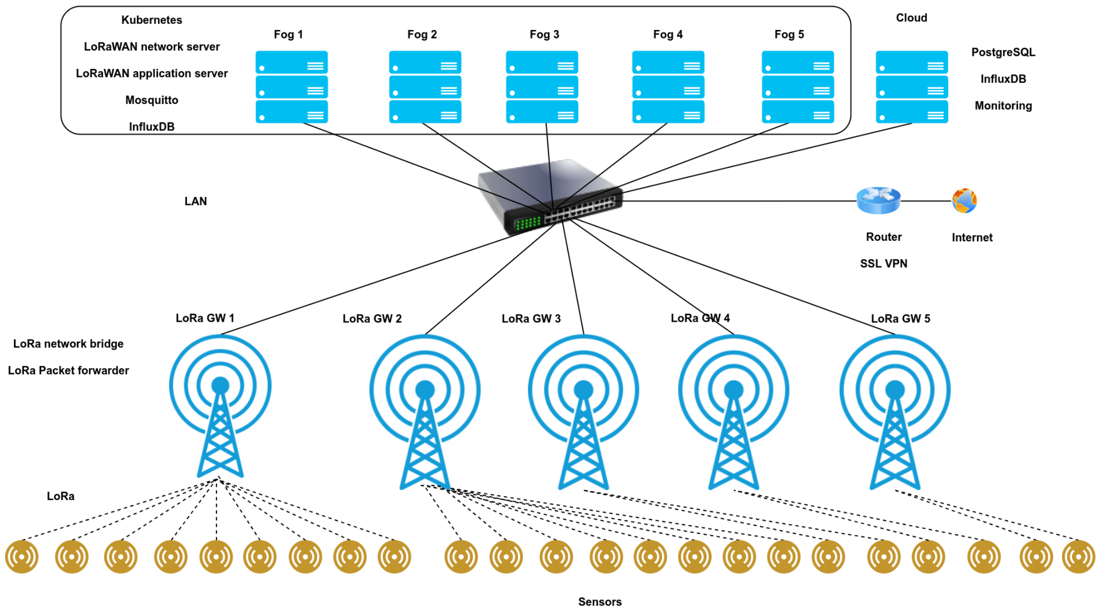

Fig. 1: High level architecture of the IoT and fog computing platform at La Marina de València.

Thirdly, the FogGuru researchers aim to address some research challenges with the IoT and fog computing platform such as:

- Support for multiple sensors
- Resource management and application deployment in multiple fog computing clusters
- Multi-tenancy and security
- Data decoding and pre-processing

B. Selection of sensors

We selected the sensors shown in Table I based on the parameters that the marina wanted to be measured. The criteria for selecting the exact sensors was based on the fact that we are building up on and extending the LivingFog platform. As a consequence, all of the sensors except the sea wave sensor are based on LoRaWAN technology. As for the sea wave sensor, 4G was used as the networking technology to allow the transfer of large amounts of data that the sensor generates to the fog computing platform.

C. Sensor network

In order to forward the data generated by the sensors over LoRa to the fog computing platform, we use LoRaWAN gateways. We have selected two outdoor and three indoor gateways placed at different locations inside La Marina de València to forward the data from the sensors in their vicinities. The list of LoRa gateways used in this project is shown in Table II.

D. Fog platform hardware

Our fog computing platform is based on the Picocluster 5 and Picocluster 10 which are 5-node and 10-node clusters based on the Raspberry Pi 4B. These single-board computers are small and powerful enough as well as low-cost. They offer sufficient computing power and storage to store and process the sensor data. The list of fog computing clusters used in this project is shown in Table II.

E. Fog platform software

Apart from the challenges mentioned in a previous chapter, multiple software challenges were addressed in our fog platform. The platform is built using various open-source frameworks.

Various vendors and LoRaWAN devices: One design specification of creating a Fog platform was to build a platform using relatively powerful, lightweight, low-cost computers. The LivingFog platform used 60 Raspberry Pi single board computers, divided into 7 physical clusters. Plus, multiple LoRaWAN gateways and sensors had to be configured and integrated into the platform. Orchestrating a data exchange between multiple Fog computing clusters challenged us to use scalable cluster orchestration systems.

Continuous data flow from multi-vendor sensors: Continuous and consistent data flow from every sensor needed to be provided. We needed to use a common data protocol and had to design a custom data processing middleware.

Multi-tenancy and security: A fog computing platform should be used between multiple tenants. Meaning, multiple application developers must deploy their own fog applications using on the Fog platform, using the data the Fog platform provides. Also, the platform must support the secure connection and login from multiple application developers.

Resource management and scalability: The platform had to be created with scalability in mind. The Fog platform had to be scalable based on the data load and the number of tenants who are using the platform. Also, the resource management and the state of the platform itself must be monitored.

Kubernetes, an open-source container orchestration platform, is used to run multiple services on the fog clusters.

TABLE I: Type and number of sensors selected for the project as well as the parameters they measure.

No.	Sensor name	Model	Parameters	Qty
1	Sea wave and current	Nortek AWAC 1MHz	Sea wave, current, temperature, pressure	1
2	Weather station	Libelium-Gill-EX-Machina Smart Weather Forecast LoRaWAN Solution Kit	Temperature, humidity, air pressure, rain fall level, solar radiation	1
3	Wind sensor	DecentLab DL-ATM22	Wind speed, direction, air temperature	3
4	Water quality sensor	Libelium-Thing+ LoRaWAN Smart Fish Farming Solution Kit	Conductivity, pH, dissolved oxygen, water temperature, Ammonium ion, Nitrite ion	1
5	Traffic counter	Parametric TCR-LS slow traffic counter with LoRaWAN	No. of vehicles, speed	10
6	People counter	Parametric PCR2-OD outdoor radar people counter with LoRaWAN	No. of pedestrians	4
7	Indoor environment sensor	enLink Air wireless air quality sensor	Temperature, humidity, light Level, VOCs (Volatile Organic Compound), Carbon Dioxide, particulate matter (PM2.5 & PM10), Oxygen, sound level, barometric pressure	3

TABLE II: List of LoRa gateways and fog computing clusters used in the project.

No.	Item type	Model	Qty
1	Indoor LoRa gateway	Multitech Conduit MTCDT-L4E1-247A-868-EU-GB	3
2	Outdoor LoRa gateway	MultiTech Conduit MTCDTIP-L4E1-267A-868 LTE Cat 4 AEP Conduit IP67	2
3	Fog computing cluster with 5 nodes	Pico 5 Raspberry PI4 4GB + 160 GB of SD storage	2
4	Fog computing cluster with 10 nodes	Pico 10H cluster Raspberry PI4 4GB + 320GB of SD storage	5

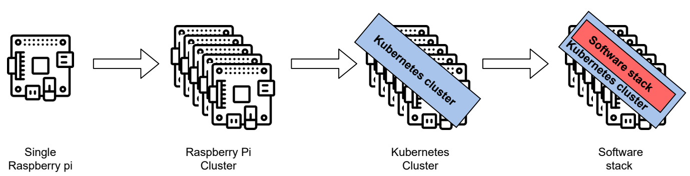

Fig. 2: Software stack layers deployed on the Fog platform

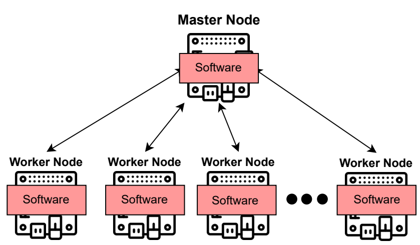

Fig. 3: Kubernetes cluster structure.

Kubernetes is a portable, scalable and extensible resource manager that allows to deploy applications declaratively. Figure 3 shows the architecture of Kubernetes.

Figure 2 shows the overview of the configuration steps needed for application-ready Fog node. After creating the Kubernetes cluster, various software and services were deployed on a cluster to create an actual Fog platform.

Figure 4 shows the combination of multiple volumes into one volume using GlusterFS. GlusterFS (Gluster File System) is an open source Distributed File System that can scale out in building-block fashion to store multiple petabytes of data. The clustered file system pools storage servers over TCP/IP or InfiniBand Remote Direct Memory Access (RDMA), aggregating disk and memory and facilitating the centralized management of data through a unified global namespace. The software works with low-cost commodity computers and is based on Linux. Using GlusterFS, we were able to combine 5-

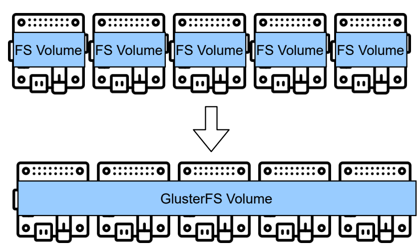

Fig. 4: Data storage using the GlusterFS scalable network filesystem.

10 separate storages from multiple Raspberry Pis, ranges from 32GB to 64GB into one single 160GB to 640GB filesystem storage. This solution was needed when the data collected from multiple sensors grew periodically.

Figure 5 shows the relation between software stacks and services installed and configured on each cluster.

- Chirpstack Application Server - The ChirpStack Application Server is a LoRaWAN Application Server, compatible with the ChirpStack Network Server. It provides a web-interface and APIs for management of users, organizations, applications, gateways and devices. The received uplink data is forwarded to one or multiple configured integrations.
- Chirpstack Network Server - The ChirpStack Network Server is a LoRaWAN Network Server, responsible for managing the state of the network. It has knowledge of device activations on the network and is able to handle join-requests when devices want to join the network. When data is received by multiple gateways, the ChirpStack Network Server will de-duplicate this data and forward it as one payload to the ChirpStack Application Server. When an application-server needs to send data back to a device, the ChirpStack Network Server will keep these items in queue, until it is able to send to one

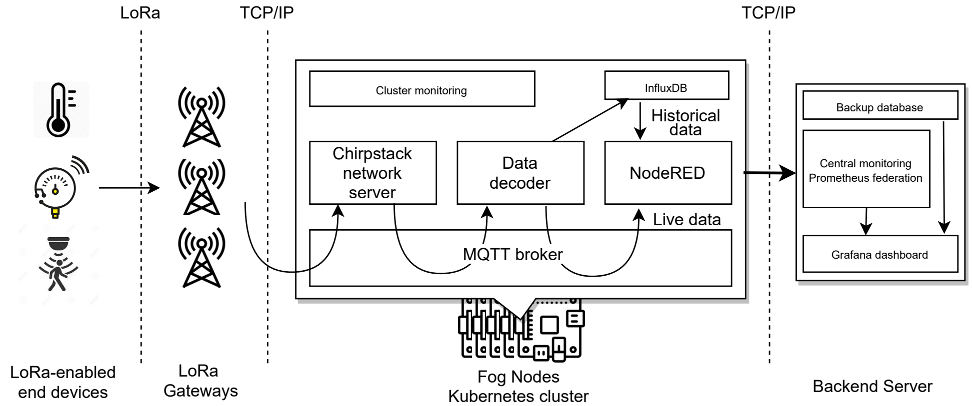

Fig. 5: Software stack and its data stream

of the gateways.

- Chirpstack Gateway Bridge - The ChirpStack Gateway Bridge sits between the Packet Forwarder and MQTT broker. It transforms the Packet Forwarder format (like the Semtech UDP Packet Forwarder protocol) into a data-format used by the ChirpStack components. The Gateway Bridge is not installed on RPi clusters. Rather, it is installed and configured on each LoRa gateway device.
- MQTT broker - MQTT is a publish/subscribe messaging system. In our system, it acts as a main messaging system, it is in control of handling all the major data flow between Chirpstack Gateway Bridge, Chirpstack Application and Network Servers, and data parser python code.
- PostgreSQL - Acts as a main database for storing information about the devices connected to the Chirpstack Application server.
- Redis - ChirpStack Network Server uses Redis for storing device-session data and non-persistent data like distributed locks, deduplication sets and meta-data.
- Data management middleware - Its purpose is to work on incoming data from Chirpstack MQTT, interpret them according to the dataframe of each sensor, and forward them back into MQTT broker and InfluxDB. A more detailed explanation of the data management middleware is given in the section below.
- InfluxDB - We use InfluxDB to store historical measurements data coming from the sensors.

F. Data Management Middleware

IoT devices, such as sensors, are an essential part of a fog-enabled IoT platform. These devices are the sources of data that provide abundant information about the monitoring environment and/or real-time events (e.g., the number of passing cars or pedestrians) , which can be used for application development. IoT devices deployed in IoT environments are from a variety of vendors and tend to produce a large volume of data on a daily basis [12]. To manage the data well, some requirements emerge for the fog platform, which are described as follows:

- The heterogeneous data sources require fog platforms to manage the devices uniformly and make all the sensor data interoperable for applications
- The vast volume of data requires high processing and storage capacity of the platform, among which data preprocessing is necessary to improve performance and to save bandwidth of data transmission. Moreover, both centralized and decentralized data storage are required to improve application performance.
- The dynamic nature of IoT devices (e.g., adding them to a network in a flexible manner) requires fog platforms to be able to discover the devices without any manual interventions.

To meet the above requirements, our LivingFog platform includes a data management middleware that can holistically manage the massive volume of data from various IoT devices. It supports device discovery and enables the development of batch and stream processing applications using real-time and historical data. In addition, this middleware can be deployed as an individual service in Kubernetes, making it scalable to the number of fog nodes.

Figure 6 gives an overview of our data management middleware. There are two inputs: sensor data from the MQTT broker (encapsulated in LoRaWAN format) and customized sensor information, including the device identifiers, sensor parameters, and LoRaWAN protocols.

The middleware subscribes to the MQTT broker via topics and processes the data stream received from sensors. The data from the MQTT broker are decoded and formatted with the LoRaWAN protocol of the sending sensor.

After receiving the data, the middleware recognizes the sensor type by matching the device identifiers and reads the measured data using the LoRaWAN protocols provided in the input. Lastly, the readable sensor data are sent to a local database running in the fog cluster and also forwarded to a global database in the centralized cloud.

G. Monitoring

Monitoring is critical for a fog platform to guarantee the availability and reliability of the applications and resources.

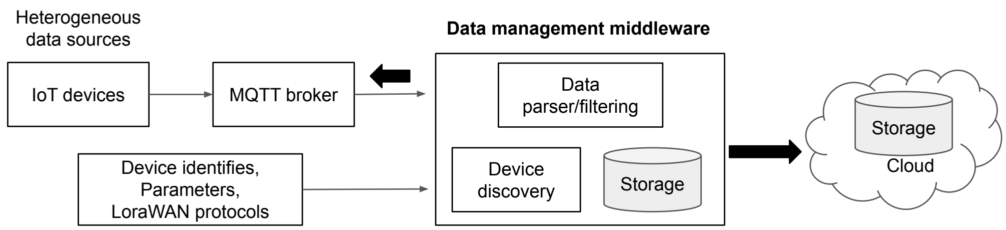

Fig. 6: Data management middleware in the LivingFog Platform.

For instance, resource metrics can be used to detect over- or under-utilization of resources [13], [14]. In addition, the observational metrics provided by a monitoring system also play a vital role in providing advanced services such as scheduling, scaling and migration.

To this end, our Livingfog platform is equipped with a monitoring stack that tracks and collects information about computing resources, the status of computing devices, and the performance of deployed services (e.g., network server in Chirpstack). This monitoring stack is composed of a set of tools, including Node-exporters[1] for resource utilization of fog nodes, kube-state-metrics[2] for the status of the Kubernetes orchestration platform, cAdvisor[3] for resource utilization of containers on which services run. All the data collected by the above tools are centralized in Prometheus[4], where the metrics can be queried, and from which alerts for unexpected events can be sent if so configured. Moreover, Grafana[5] is used for visualizing the data stored in Prometheus.

H. Security features of the platform

One of the objectives of the LivingFog platform is to enable a hackathon and IoT fablab so that citizens can make use of the data generated by the sensors. In this objective, citizens who are interested in developing fog computing applications should be able to access our fog platform and deploy their application on it. Therefore, we need to define the users' access level to the resources of the platform in such a way as to provide enough freedom for the application developer, and at the same time to prevent them from any unauthorized access to other users' or the platform's resources. During the hackathon, the importance of the separation of the users in the form of teams and also the protection of their assets increases.

As our fog platform is based on Kubernetes, we use Kubernetes' role-based access control (RBAC) mechanism to define the roles and their corresponding access level. When defining a role, we specify the resources that each role has access to them and the actions that can be performed on that resource using the role. After defining roles, we generate a token that grants access to the platform concerning the defined role in the team namespace.

[1]Node-exporter - https://github.com/prometheus/node_exporter
[2]kube-state-metrics - https://github.com/kubernetes/kube-state-metrics
[3]cAdvisor - https://github.com/google/cadvisor
[4]Prometheus - https://prometheus.io/
[5]Grafana - https://grafana.com

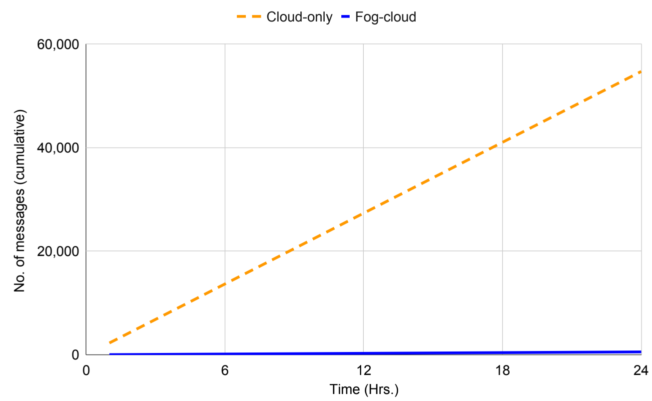

Fig. 7: Cumulative number of messages for a day.

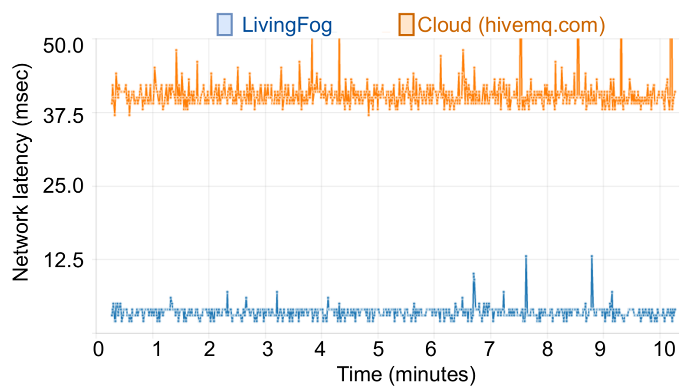

Fig. 8: Latency (msec) of publishing to MQTT topics.

The fog nodes have a limited amount of CPU and memory resources. As our fog platform hosts multiple applications from different teams simultaneously, we anticipated a condition where some hackathon teams exhaust the platform resources and disturb other teams. To avoid such conditions, we exploited the Kubernetes resource quota. This mechanism in Kubernetes allows us to define a quota for the said resources at the container level. However, we defined some default values for the CPU and memory usage by each container to facilitate the use of our platform.

IV. Evaluation

In this section, we present the performance of the LivingFog platform as compared to a traditional cloud-only deployment.

We use the number of messages sent over the network and network latency as metrics to compare the two deployments.

Figure 7 shows the number of cumulative messages sent by the sensors deployed at La Marina de València during a day. In a traditional cloud-only deployment, all messages from all sensors should be sent directly to the cloud platform over long distances. As a result, we see a large number of messages sent amounting to a cumulative of 54,720 messages per day. In contrast, in the case of the fog-cloud deployment scenario, sensor messages are sent directly to the fog platform for pre-processing and only messages containing aggregate values are sent to the cloud every hour for visualization and persistent storage. In the latter case, a total of 552 messages are sent to the cloud during a period of 24 hours.

Figure 8 shows the network latency for publishing sensor messages to MQTT brokers in the fog and cloud platforms. We see that publishing messages to the MQTT broker on the fog platform takes on average only 4 ms, whereas sending messages from the sensors to an MQTT broker in the cloud (hosted in hivemq.com in this case) takes on average 40 ms. Clearly, this shows the advantage of fog deployments in reducing network latency significantly.

V. Use cases

In this section, we describe the top three use cases introduced by teams of the "Hack the Fog!" hackathon[6]. The two main parameters for selecting the winner use case were: First, The usefulness of the use case, and second, the level of technology that the team used for prototyping their applications. Table III represents the top three selected use cases.

The first place went to "Fog pandits" team. They target the water siltation problem. Siltation is the water pollution caused by particulate terrestrial clastic material, with a particle size dominated by silt or clay [15]. Water siltation prevents the passage of vessels and decreases water quality and biodiversity. The team's approach is to use the sensors data to model the siltation in water and notify La Marina with the exact location where the level of siltation is above the normal level. The said team utilized the data received by sea wave, current profiler, and water quality sensors to feed the stiltation model. They deployed their application using Node-RED[7] for processing the streams of sensors data in real-time.

The second winning team was "Marina Connect". The team developed an application for both the restaurant owners in La Marina area and their customers. The restaurants located in the La Marina area may face decrease in the number of their customers due to pandemics or bad weather conditions. On the other hand, customers prefer to go to the restaurants knowing that the place is not crowded in Covid pandemic situations[8]. The *Marina Connect* application uses real time sensor data of people counter, car counter, weather station sensors (air temperature and precipitation data) and wind sensors along with the historical data of these sensors stored in the cloud to predict the people congest in the La Marina area. Based on this information, the application sends incentives to the potential restaurants' customers to enjoy attending the restaurants.

The third winning application of the hackathon was "Hodor". The application targets a similar problem to the previous application, hence with a wider range of audiences. This application uses sensors data to notify people if they can attend different spots of La Marina without risking their health in pandemic situations. The application takes into account the data of people counter sensors, wind sensors, weather station sensor and water temperature sensors to decide whether it is safe for an individual to enter an spot in La Marina.

[6]The hackathon, in which 63 participants used the LivingFog platform and the data collected from the sensors to validate their ideas, took place from 26 to 28 March, 2021 in Valencia and fully online from 26 to 28 March, 2021 https://ec.europa.eu/info/events/hack-fog-2021-mar-26_en.

[7]Node-RED - https://nodered.org/

[8]Crowding is one of the main factors in increasing the risk of respiratory viral infections [16].

VI. Conclusion

In this paper, we present an experience report of the LivingFog fog computing platform integrated with LoRaWAN technology that has been used for a smart marina project at La Marina de València, Spain. In this project, several sensors were deployed inside La Marina to measure various parameters of the sea, weather, movement of people and vehicles. The data collected from the sensors is processed on the fog computing platforms. We report on the performance of the platform in terms of latency reduction and saving of bandwidth. Moreover, the platform has been used to deploy various applications by participants of the "Hack the fog" hackathon to validate their innovative ideas. The platform is available under a liberal open-source licence[9]. We hope this paper encourages other researchers to report on similar fog platforms.

[9]LivingFog - http://www.fogguru.eu/livingfog/

References

[1] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges, and optimization," *IEEE Access*, vol. 5, 2017.

[2] H. Arkian, D. Giouroukis, P. S. Junior, and G. Pierre, "Potable water management with integrated fog computing and lorawan technologies," *IEEE IoT Newsletter*, 2020.

[3] S. Li, L. Da Xu, and S. Zhao, "The internet of things: A survey," *Information Systems Frontiers*, vol. 17, no. 2, 2015.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, 2010.

[5] D. M. Mendez, I. Papapanagiotou, and B. Yang, "Internet of things: Survey on security and privacy," *CoRR*, vol. abs/1707.01879, 2017.

[6] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog computing for the internet of things: A survey," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 2, 2019.

TABLE III: The top three use cases of the hackathon teams.

	Sensors data	Fog used for	Is cloud used?
Detect and alarm the water siltation	Sea wave, current profiler, water quality	Processing	No
Predict a calm day in a restaurant and offer incentives	People counter, car counter, weather station, wind	Processing	Yes
Inform if a spot is safe in pandemic situations	People counter, wind, weather station, water temperature	Processing	No

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC Workshop on Mobile Cloud Computing*, 2012.

[8] A. Ahmed, H. Arkian, D. Battulga, A. J. Fahs, M. Farhadi, D. Giouroukis, A. Gougeon, F. O. Gutierrez, G. Pierre, P. R. Souza Jr, M. A. Tamiru, and L. Wu, "Fog computing applications: Taxonomy and requirements," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1907.11621.

[9] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled iot networks," *IEEE Communications Magazine*, vol. 55, no. 8, 2017.

[10] S. A. Noghabi, L. Cox, S. Agarwal, and G. Ananthanarayanan, "The emerging landscape of edge computing," *GetMobile: Mobile Computing and Communications*, vol. 23, no. 4, 2020.

[11] M. Yannuzzi, F. van Lingen, A. Jain, O. L. Parellada, M. M. Flores, D. Carrera, J. L. Pérez, D. Montero, P. Chacin, A. Corsaro, *et al.*, "A new era for cities with fog computing," *IEEE Internet Computing*, vol. 21, no. 2, 2017.

[12] M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an iot-based industrial data management for smart manufacturing," *Journal of Sensor and Actuator Networks*, vol. 8, no. 2, 2019.

[13] S. K. Battula, S. Garg, J. Montgomery, and B. Kang, "An efficient resource monitoring service for fog computing environments," *IEEE Transactions on Services Computing*, vol. 13, no. 4, 2019.

[14] V. K. Prasad, M. D. Bhavsar, and S. Tanwar, "Influence of montoring: Fog and edge computing," *Scalable Computing: Practice and Experience*, vol. 20, no. 2, 2019.

[15] *Siltation*, https://en.wikipedia.org/wiki/Siltation.

[16] B. Rader, S. V. Scarpino, A. Nande, A. L. Hill, B. Adlam, R. C. Reiner, D. M. Pigott, B. Gutierrez, A. E. Zarebski, M. Shrestha, *et al.*, "Crowding and the shape of covid-19 epidemics," *Nature medicine*, vol. 26, no. 12, 2020.