

Essentiality of Managing the Resource Information in the Coordinated Fog-to-Cloud Paradigm

Souvik Sengupta | Jordi Garcia | Xavi Masip-Bruin

Dept. of Computer Architecture (DAC), UPC-BarcelonaTech, Vilanova i La Geltrú, Barcelona, 08800, Spain

¹Dept. of Computer Architecture (DAC), UPC-BarcelonaTech, Vilanova i La Geltrú, Barcelona, 08800, Spain

Correspondence

Souvik Sengupta, Dept. of Computer Architecture (DAC), UPC-BarcelonaTech, Vilanova i La Geltrú, Barcelona, 08800, Spain
Email: souvik@ac.upc.edu

Funding information

This work has been supported by the Spanish Ministry of Science, Innovation and Universities and by the European Regional Development Fund (FEDER) under contract RTI2018-094532-B-I00, and by the H2020 European Union mF2C project with reference 730929.

Fog-to-Cloud (F2C) computing is an emerging computational platform. By combining the cloud, fog, and IoT, it provides an excellent framework for managing and coordinating the resources in any smart computing domain. Efficient management of these kinds of diverse resources is one of the critical tasks in the F2C system. Also, it must be considered that different types of services are offered by any smart system. So, before managing these resources and enabling the various types of services, it is essential to have some comprehensive informational catalogue of resources and services. Hence, after identifying the resource and service-task taxonomy, our main aim in this paper is finding out a solution for properly organizing this information over the F2C system. For that purpose, we are proposing a modified F2C framework where all the information is distributively stored near to the edge of the network. Finally, by presenting some experimental results, we evaluate and validate the performance of our proposing framework.

KEYWORDS

Fog, Cloud, IoT, F2C, Taxonomy, Resources Management

1 | INTRODUCTION

In modern days, technology has the most significant impact on society; it cannot be separated from our life. We are using technology to ease our daily lives; it helps us to improve the quality of living. Thus, to fulfil our needs and demands, technology keeps on rising. We are using the technology for travelling, communicating, learning, doing business and, mostly in-everywhere to improve the comforts of our living. As a result, it is always changing as more inventions are forwarding to improve our daily lives.

As a consequence, the whole of society is moving towards *smarter* world. As a part of the ever-evolving technology, information and communication domains are also experiencing fast and rapid developments. The modern information and communication technologies are more pervasive, more efficient, more agile and having high computational capabilities than the old versions. Significantly, all of our surrounding things are gradually getting connected to the network.

The preliminary objective of this development is to provide better and real-time services to the human. For achieving this goal, in [1], the authors proposed a new emerging technological solution by combining the cloud computing, fog computing and IoT concepts, which is known as the Fog-to-Cloud (F2C) computing. As per them, for providing exceptional service facilities in smart environments, the F2C has emerged. In their work, they introduced the F2C platform as a combined and hierarchical computing framework. According to them, F2C helps to optimize resource utilization and improve service execution by optimally mapping services into the resources, according to the best matched the services demands.

The F2C still is in the infantile state, and it is experiencing the development. The authors in [1] considered the F2C is an emerging platform, and they said that the F2C might helps the smart computing system to be much smarter. Most importantly being a developing platform, by addressing lots of existing challenges in the smart paradigm, it has to come up with proper solutions for providing better services. For that reason, it is necessary to define and validate the proper resource management strategy in the F2C platform. Considering various smart computing paradigm, we understand that, the enormous diversity of the system resources and smart system services is creating a huge problem to build a proper resource management mechanism in that domain. So, considering this fact, it is necessary to define the system resource classification and also to identify the system-involving service requirements in any F2C-enabled smart system.

Interestingly, all of these kinds of information eases the process for matching the system resources with the requested service requirements. So before building up the proper resource allocation and management schema in the F2C enabled smart system, it is essential to continuously monitor the participating system resources for knowing their availability and capability. Besides that, any smart system (i.e., smart city, smart industry, smart farming) consists of a massive amount of sensor nodes [2], which are continuously capturing the data from various environmental events. So, along with the sensed data, this kind of continuously captured resource information creates a massive data flow over the F2C enabled smart system. So, managing this enormous amount of information creates a massive challenge for any F2C enabled smart environment [3].

In this paper, considering this fact, we are focusing on identifying a solution for managing this endless amount of monitoring information over the system for build up a proper resource management schema in the coordinate F2C paradigm. Initially, in this paper, we briefly present our taxonomic model for identifying the characteristics of the F2C system resources and involved service-task and also discuss the monitoring module to keep track of the availability of system resources. In our ongoing project [4], we are using this taxonomic model and the resource-monitoring module for properly managing participating devices in various use-cases where a resource-monitoring model involves for continuous collection of available system resources' information. Then, for properly manage this information and

to build the system more agile, we are proposing a modified F2C framework where we consider a distributed database cluster near to the edge of the network architecture. Finally, by presenting some experimental results, we argued the performance of our proposed framework.

The rest of the paper is organized as follows. Following the F2C architectural view in a smart environment system, we briefly discuss the resource management strategy in Section 2. After that, in Section 3, we follow various related works. Then in Section 4, we identify all critical issues for the necessity of managing this information, in order to define the proper resource management strategy in the F2C paradigm. In Section 5, by briefly focus on the taxonomic models and the resource monitoring mechanism, we describe and propose a framework for managing the captured information over the F2C system. After that, in Section 6, we presented some experimental results in order to justify the efficiency of the proposed framework. Then finally, by introducing future directions, we present the concluding remarks of our research work in Section 7.

2 | THE F2C RESOURCE MANAGEMENT FRAMEWORK

According to the author in [1], the F2C has appeared for bringing more smartness to any smart computing paradigm. By securing the implementation of advanced services and better resource utilization mechanism, the F2C promised to bring more intelligence to any smart system. For better realizing the assimilation of the F2C and any smart environment scenario, in Fig. 1 (i), we illustrate an F2C enabled smart city scenario. As per the authors, the F2C platform is a combined, layered and hierarchical computing platform (as shown in Fig. 1 (ii)), where all the cloud resources reside on the top layer, and the bottom layer consists of a massive amount of IoT devices. According to the hierarchical architecture, fog resources reside between the cloud layer and the IoT layer. Significantly, by following these two diagrams, it is easily recognizable that, within the F2C-enabled smart city, multiple numbers of Fog Areas (FA) can appear. Where for each FA, a computing node is responsible for providing the F2C facilities, which is known as the Fog Leader node (FL). Interestingly, in each FA, a vast number of heterogeneous IoT devices and other computing devices are participating.

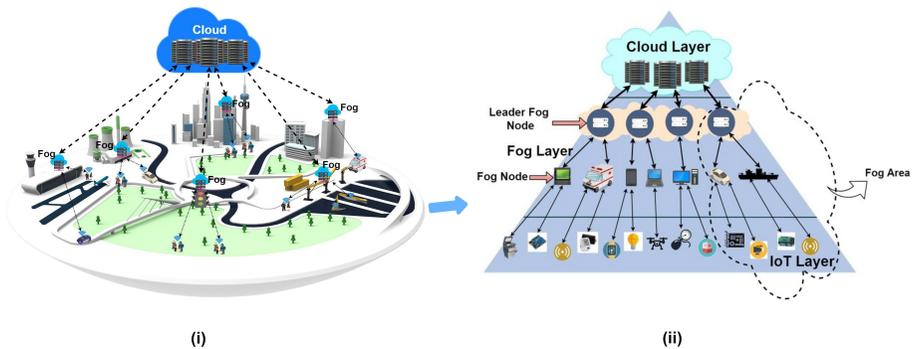


FIGURE 1 (i) The Fog-to-Cloud enabled smart city scenario; (ii) Hierarchical architecture of the Fog-to-Cloud.

Also, by analyzing the smart city scenario and these two diagrams, it can be easily determined that the diverse nature of the considering system is the crucial issue. Significantly, the diverse nature of participating resources and the variety of system involved services creates massive challenges for managing them. Interestingly, the services which are involved in any smart computing scenario have different resource requirements. Most importantly, that need to be

satisfied before services to be delivered among the smart city's resident. However, it is quite tough work to confirm that all of these requests must satisfactorily regulate and build an efficient system. Thus considering these facts, in the same paper [1], the authors realized for having an exhaustive resource management mechanism to develop an efficient F2C system and manage the system resources. But, without knowing the resource characteristics, it is quite impossible for managing the system resources. Significantly, considering the smart city scenario, we understand that, to adequately manage the system resources, along with the resource characteristics information, it is also necessary to continuously collect their present quantitative information. Also, for improving the system functionalities, it is necessary to figure out the features and requirements of the system involved tasks and services.

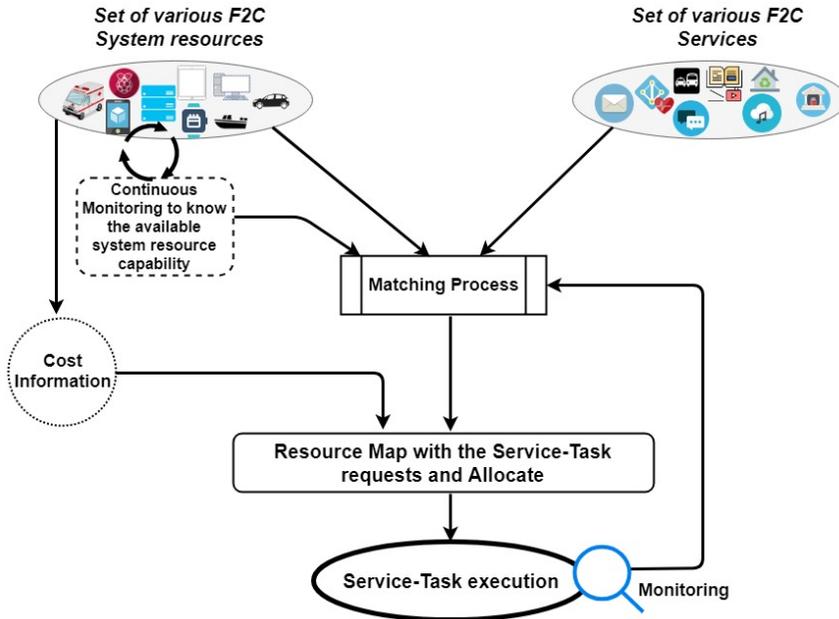


FIGURE 2 Resource management strategy in the F2C paradigm.

Considering, the F2C platform and any smart city scenario, we understand that with the help of the aforementioned information and by knowing the user preferences, the appropriate resources can be allocated for executing some task(s) and serve the service(s) among the smart city's resident(s). Significantly, the 'cost' has a considerable influence to choose the appropriate resources and most importantly, it helps to define and design the proper and cost-efficient resource management mechanism in any service-oriented system (i.e., smart city application). To better understanding the resource management mechanism in the F2C domain, we depicted the resource management strategy in Fig. 2. Significantly, we found that the 'cost' not only depends on the system resources specifications but, implicitly it also depends on the service-task requirements [5, 6]. The real fact is that, without having this information, it is a hard job to calculate the 'cost' for services or even, define the 'cost-model' of a system. So, it is clear that this information is really vital for any efficient computing platform. In addition, it needs to be properly managed and stored over the computing platform. By inspecting the past related works, in the next section, we are going to figure out how these classification or characterization information and their proper organization are helping to design the proper resource management strategy in the coordinated F2C paradigm.

3 | STATE OF THE ART: RELATED WORK

Build up a proper resource management strategy is a crucial issue for any service-based system (i.e., smart city application). By reviewing several related works, we found that it is necessary to determine the characteristics of participating system resources and also the service execution requirements, before making a proper resource management strategy. Also, we found that to build a more sophisticated and more efficient resource management mechanism; it is essential to continuously monitor the available system resources and organize it accurately over the system. So, following various past related works, in this section, we understand how the resource and service-task characteristics information helping to design the proper resource management strategy for their considering system. Also, by considering various works, we aim to know that, how the proper management of this information helping their system to be more efficient.

To build the sustainable sensing infrastructure for smart city applications, in [7], authors presented an extensive survey to know the characteristics of their considering system resources. Most importantly, in that paper, the authors argued the fog computing platforms helps to build a viable sensing infrastructure for any smart city applications. According to authors in [7], fog computing is still in the infantile stage and is an emerging technology. So to build an efficient fog computing platform, it is necessary to have some comprehensive, proper and efficient resource management mechanism. Accounting this fact, in [8], the authors presented a strategy for dynamic resource allocation in fog computing.

Most importantly, before presenting the allocation strategy, they identify the characteristics of their system resources and task by adopting the Preemptive Time Petri Nets (PTPN) model. Similarly, in another paper [9], authors presented a resource management framework model in the combined fog-IoT based system; where they considered the characteristics of their system resources for presenting resource usage prediction, resource estimation and reservation model. Also, based on the features of their system resources, they calculate the cost for their system resources. Also, in [10], the authors presented that, how the cloud resource description and continuously monitoring the system resources, help to design better resource utilization and load-balancing in the Cloud-NFV platform. Significantly, task scheduling is a vital aspect of any distributed computing platform [11]. So, considering this fact in [11], the researchers presented a methodology for scheduling the task in the diversified distributed computing paradigm. In that paper, they said that the distinct nature of the system resources and tasks have a significant impact on building the task scheduling model for their considering system. Likewise, in [12], by presenting an extensive literature survey, the authors understood that before defining the adequate resources management mechanism, it is necessary to know the characteristics of the cloud resources.

For other computing platforms, we also found that researchers analyzed their system resources and service-task characteristics to define the suitable resource management strategy. For example, in [13], the authors did an extensive literature survey for the OSs that could become the standard OS for low-end IoT devices and then in that paper, they classified the IoT resources based on their capabilities, so that helps to build an efficient IoT enabled system. Likewise, in [14], by identifying the characteristics of the system resources and also the services, the authors presented the resource management mechanism in the smart city domain. Furthermore, in some research works ([15], [16]), based on the characteristics information of the system resources and identified service-task demands, the cost model of the corresponding computing paradigm has been defined. Most importantly, in some work ([17]), the cost model has a huge impact on defining an efficient resource management mechanism in the various computing platform.

Besides all of these aspects, unfortunately, only the resource and service-task characterization information are not sufficient to design a proper and efficient resource management mechanism. Considering this fact, in [18], the researchers have proposed an architectural model for monitoring their system resources in fog and mobile cloud platforms. In that paper, the authors said that efficient system resources monitoring is necessary, and it is also essential

to have an appropriate distributed mechanism for storing and collecting these monitored resources information in their considered system. According to the authors [18], that could help their system to be more agile and provide better latency-sensitive services. To a better realization of the data collection mechanism in the distributed computing system, authors in [19] present an extensive study for efficiently collecting the data in the VANET platform.

After reviewing these works, it is undoubtedly clear that along with the resource and service-task characterization models, it is also necessary to build a proper mechanism for monitoring the system resources. Additionally, along with the captured sensed data, the monitoring information is generating a tremendous amount of data flow over the system. So, building a proper data storing and managing mechanism is necessary before setting up an adequate resource management technique in the F2C-enabled smart computing domain. However, unfortunately, we have not found any work where researchers present a solution for distributedly managing the monitoring resource information over the network to build an efficient resource management mechanism in a hierarchical distributed computing paradigm, such as F2C or similar. For this reason, that becomes one of the key influences for proposing the distributed database based on the F2C paradigm.

4 | PROBLEM STATEMENT: ESSENTIALITY OF THE RESOURCE AND SERVICE-TASK CHARACTERIZATION

Focusing on the F2C-enabled smart city scenario and following all the related works, it is easily understandable that immense diversification is one of the most vital aspects of this newly rising, layered, and hierarchical architecture-based computing platform. In the case of both resource and service parts, we identified that the F2C computing platform has full of divergence. For example, a massive number of devices are participating in this computing paradigm. Significantly each of them having different specs and also they are different in nature. Similarly, various services are offered in a smart city platform. Fundamentally, every service has different requirements. For, fairly provide the service(s) among the smart city's citizen, it is necessary to satisfy all these demands. As we earlier found that, without having the proper knowledge about the characteristics of participating resources and service-demands, it is truly a strenuous work to organize the F2C system resources properly. Also in earlier, we recognized that, for any service-based system, an appropriate cost-model is playing a crucial role to define and organize the adequate and cost-effective resource management mechanism. Interestingly, in ([9], [15], [16]) the authors considered the characteristics of system resources and information about service-tasks requirements for defining the cost-model of the considering platform.

By understanding these aspects, we realize the essentiality and relevancy for having such kind of characterization catalogue for both of the resources and the services, in the F2C-enabled smart system. Considering these facts, in our previous work [20] we have already presented the characterization model for the F2C system resources and the service-task which are involved in any F2C enabled smart system. Considering various aspects, such as device attributes (i.e., hardware, software, network specification, etc.), IoT and attached components (i.e., sensors, actuators, RFID tags, or additional attached device components), security and privacy aspects (i.e., device hardware security, network security and data security), cost information (i.e., chargeable device, non-chargeable device), and historical and behavioral information (i.e., participation role, mobility, life span, reliability, information of the device location, etc.), we have already classified all the F2C resources and presented their detail characterization model in the past work [20], which are depicted in Fig. 3.

Similarly, in order to understand F2C services and tasks on the previous work, we have also placed our focus to identify the characteristics of them. We presented the service characterization model (Fig. 4 (i)) considering five

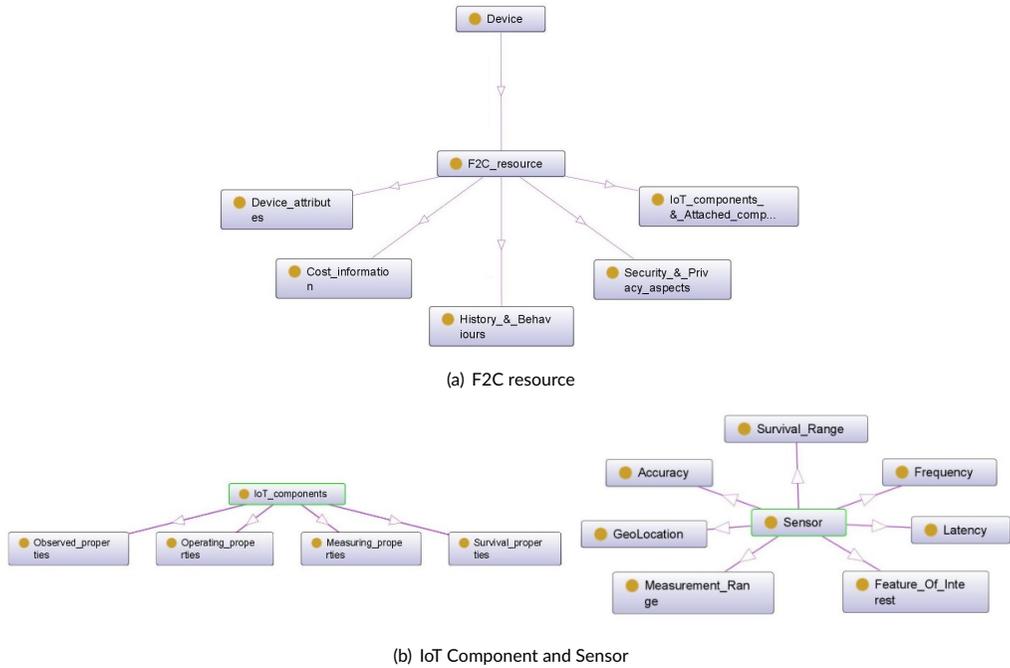


FIGURE 3 The Class diagram of Characterization Model: F2C resources, IoT Components and Sensors

different aspects: context of services (i.e., governmental, educational, transport, etc.), service location (from where the services are offered, for instance, cloud or fog), security and reliability (i.e., based on security preferences, services can be classified), data characteristics (i.e., based on the amount of data processing requirement, services can also be classified), and cost information (i.e., based on the service offering cost, services can be further classified into chargeable or non-chargeable services). Also, by considering execution requirements (i.e., network bandwidth capacity, time requirements, processor requirements, storage requirements, memory requirements, etc.) and their priority (i.e., high, medium, or low), we presented the task characterization model (Fig. 4 (ii)).

In earlier, we found that for efficiently manage the F2C resources along with the resource and service-task characteristics information, some continuous resource monitoring information is also essential. Most importantly, continuously monitoring the F2C resources means, a huge amount of information or data has to be captured. Interestingly, in [1], we found that the F2C is emerging for ensuring the improved service execution for all services including the latency-sensitive services, and also it enables better resource utilization mechanism in any smart computing platform. So, for that purpose, the F2C paradigm brings the cloud functionalities to the edge of the network and enables the opportunity to locally process the information and store it for a future purpose. Unfortunately, the devices which are participating in the fog layer are not similar to the cloud resources. The fog layer devices are mostly resources constrained. So, managing this large amount of information is quite challenging and resource-consuming for them. In addition to this, the IoT devices (mostly sensors and actuators) are continuously capturing, producing and sending data to the fog layer devices. So that creates an extra burden for the fog layer devices for managing this information. Considering all these challenges and aspects, in the next section, we are proposing a new enhanced framework for the F2C paradigm, where we are considering the distributed database over all the FL nodes to monitor and manage

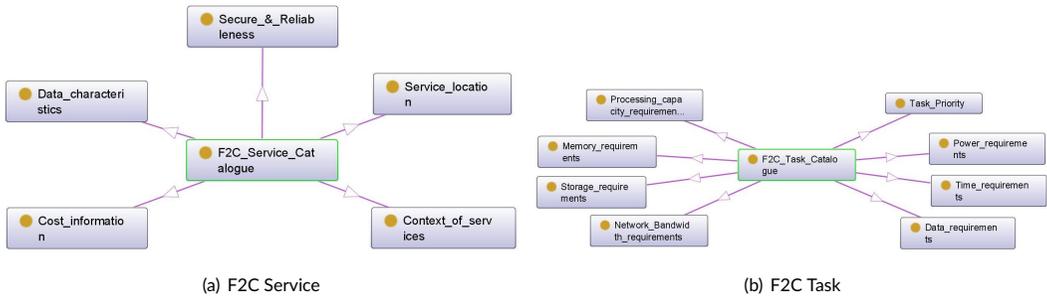


FIGURE 4 The Class diagram of Characterization Model: F2C services and F2C tasks

the system resources information; this is helping to reduce the burden of the fog layer resources and also ensures better service execution.

5 | SOLUTION: THE F2C DISTRIBUTED DATABASE FRAMEWORK

In the earlier section, by following various past works and the F2C-enabled smart city scenario, we have already identified the essentiality for properly managing the collected resource information and sensed data in the system. So considering this fact, initially in this section, we are going to briefly explain the architectural description of our proposed distributed database-based F2C platform. Then, by briefly discussing the query processing mechanism, we explain how the F2C resources can be matched and allocated for satisfying a particular requested service in the F2C domain.

5.1 | Architectural description of the F2C distributed database

As explained, the F2C is a hierarchical computing architecture where the fog layer resources reside between the cloud layer resources and IoT layer resources. Following the F2C enabled smart city scenario (shown in Fig. 1), it can be easily identified that multiple numbers of FA can exist in the F2C enabled smart city and, for each FA, a particular computing node (FL) is responsible to provide the F2C facilities. In an FA, also there might be several numbers of FN and a massive amount of sensor nodes. Note that the F2C has emerged to fill the gap of cloud computing, in case of latency-sensitive services. So it is quite essential to bring the cloud facilities near to the edge or the end-users. For that reason, we are proposing the modified F2C computing platform, where we consider a distributed database cluster over all the FLs. Fig. 5 depicts how the distributed database cluster can be implemented near to edge of the F2C paradigm. In the distributed database cluster, each FL is considered as an individual cluster node.

Following the F2C enabled smart city scenario, it is easily recognize that a huge amount of IoT devices are deployed in the smart city to capture various environmental events in real-time. For building a proper resource management mechanism in the F2C system, it is necessary to monitor the system resources continuously. Consequently, a high volume of information is generated. For efficiently providing agile or real-time services, information is required to be shared with multiple FLs. However, processing and sharing this high volume of information in real-time is a bit complicated procedure for resource-constrained FLs. Even though the cloud resources quickly enable high-level information processing, in a real case scenario, due to bandwidth limitation, it is difficult to immediately send the huge

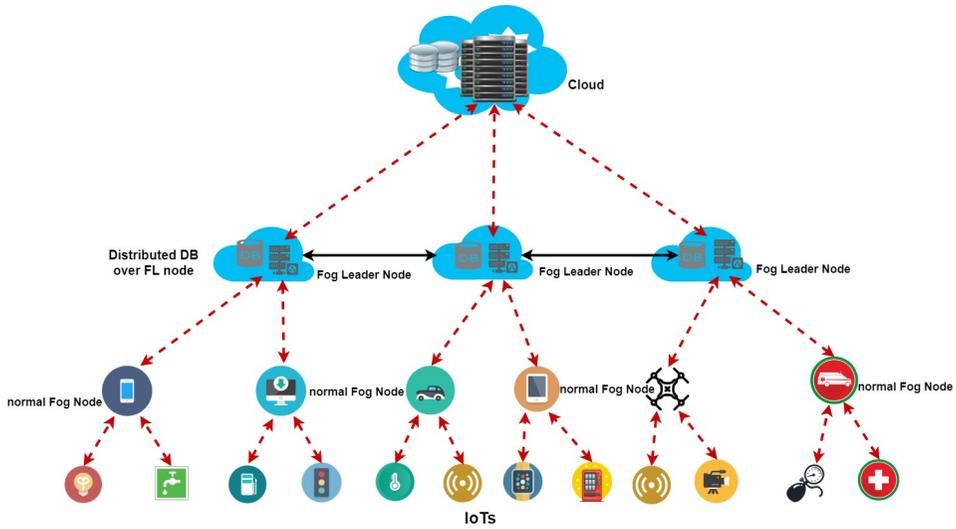


FIGURE 5 Proposed distributed database framework for the F2C architecture.

volume of collected information from the edge of the network. So, that may cause a delay to process this information and provide some decision. For example, in a latency-sensitive service (i.e., the e-Health domain or traffic management system), delay can create a huge problem. To solve that, we introduce the new F2C framework considering a distributed database over all the FLs. Although FLs are not as powerful as cloud resources, they still enable information processing near to the end-users. Thus, FLs reduce the communication burden to push information to the cloud periodically.

5.2 | Resources and service-task matching in F2C

The primary motivation of this work is to know whether the distributed data stored over the FLs are helping the system to be more agile and improve its performance. But before finding out the efficiency of our proposed framework, it is quite relevant to give some focus on identifying how the distributed database is helping the system to build a proper resource management mechanism. For that purpose, we describe all the steps which are involved in building the resource management mechanism in the F2C enabled system. Similarly, like in [21], in our framework, we also consider the student project allocation (SPA) problem to match with the available system resources and requested service requirements.

Following the F2C enabled smart city scenario, we understand that service requests can be generated in two ways. In the first case, the service requested can be generated from inside the system. That means that the user who is requesting a service is also participating in the system. In the second case, the user is not directly participating in the system; instead, it wants to access some services from outside of the F2C system remotely. In the second case, the user directly asks the cloud resources to provide the requested services, and it is out of the scope of this paper. However, for the first case, in order to satisfy the requested service requirements, the following steps must be performed:

Step 1: Search for local resources; if they do satisfy the request, then stop searching (the resources have been found); otherwise perform the next steps.

Step 2: Request to the nearest connected FL for searching the appropriate resources to satisfy the requested service requirements and, simultaneously, the requested service requirements information should be propagated over the FLs cluster.

Step 3: In the FLs cluster, using the SPA algorithm, the available resources can be grouped and classified into three categories: highly match, moderate match, and low match, as:

$$(\text{matching_score} : \gamma_r = [3, 2, 1])$$

Step 4: Considering the cost information and matching score, calculate the ranking for each resource, as:

$$(\text{resource_rank} : R_r = (\gamma_r, c_r))$$

Step 5: Based on the ranking, the appropriate resource must be chosen for the mapping with the requested service.

To build the cost-optimal efficient resource mapping, the value of the matching score (γ_r) should be high, and the cost of the resource (c_r) should be low.

Step 6: If the requested service requirements are not satisfied, then send the request to the upper cloud layer.

In this work, our sole interest is to know how the distributed database over the FLs improve the searching time for finding out a proper resource and service matching. The validation and measuring of the efficiency of the proposed resource mapping schema are out of scope for this work, and we consider it as our future work.

6 | EXPERIMENTAL RESULTS

To evaluate the performance of our proposed distributed database framework, we set up our experimental platform. After that, we perform two different tests for evaluating the performance of our proposed model. To argue the performance evaluation, we also perform the same tests in a traditional framework where we store all the information in the cloud. By comparing the performance results of our proposed model and the traditional cloud-storage based F2C, we demonstrate the effectiveness of our proposed model.

6.1 | Experimental set-up

Initially, to perform the tests, we set up the architecture by using VMs. In a physical Linux machine, which has 32GB DDR3 RAM, Intel i7 8th Gen. 8850H processor (clock speed @2.6GHz), 1TB HDD, where we have mounted three VM for the three FL nodes of three different FAs. We assigned in such a way that each FL has 4GB RAM, 50GB storage, and each of them has 2CPU cores. Also, we have mounted nine more VMs for the nine FNs in another single physical machine. In our simulated architecture, each FA has three FNs and one FL nodes. Finally, we mounted the cloud layer into a server machine which has Intel Xeon family E5-2620 V4 series (clock speed @3GHz) processor, 96GB RAM, 1TB Hard Drive running on Ubuntu 16.04LTS Linux. To keep up with the real case scenario, we simulated the F2C platform in such a way so that the network connection between the edge devices and the cloud resources has some bandwidth constraint and delay. After setting up the network architecture, we build a single datacenter and multiple

racks based on the Cassandra cluster ([22]) over all the FLs. In order to replicate the information over the Cassandra cluster, we use the *Simple Strategy* replication mechanism. Then, for the validation of our model, we perform the test on a various amount of distinct data packets. Individually the average size of each data packet is 24KB.

6.2 | Performance evaluation

To compare the performance of our proposed model to a traditional cloud storage-based framework, we perform several numbers of test for storing the data or information into the database. In Fig. 6, we presented all the test results for storing the data into the database graphically. In that graph, the blue line shows the performance of our proposed model and the red line shows the performance of the traditional cloud storage-based framework. The cloud resides far from the edge, so it is quite evident that the communication between cloud and edge must have some delay and bandwidth constraints. So that is the main reason for taking a higher time to store the data into the cloud. To deeply evaluate the performance, by considering the resource-constrained nature of the FLs, we perform the validation test on 0.128 million distinct data packets for both cases. From Fig. 6, it can be quickly concluded that considering the distributed database cluster over the FLs improves the performance of existing F2C paradigm. Though all the FLs are resource-constrained, they reside near to the edge of the network, and the Cassandra uses the consistent hashing algorithm to distribute information, which gives them a massive advantage for quickly storing the information over the cluster. Then finally, by performing the searching tests, we justified the necessity for having the distributed database over all the FLs.

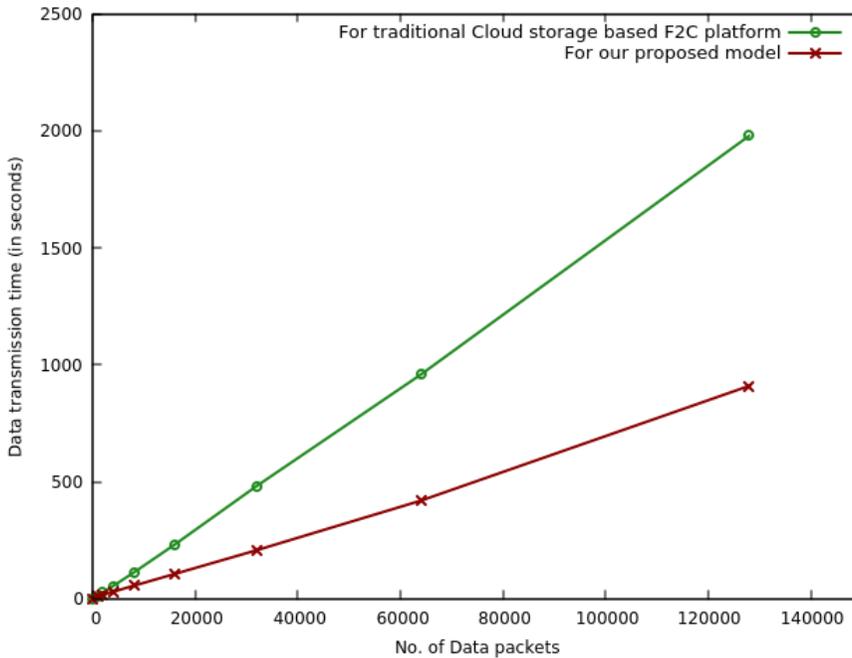


FIGURE 6 Information storing time: Traditional cloud storage-based vs our proposed model.

To measure the performance of our proposed model - considering the various number of distinct data packets,

we performed the searching tests for search the appropriate resource information to match the requested service requirements. Initially, we started the searching examination with ten thousand distinct data packets, and we ended-up our study by performing the test on 1.28 million data packets. So, initially, when we conducted our analysis for our proposed model; we found that after completing the test on a certain amount of data packets, the query response time becomes more identical. More-precisely, after reaching more than 6,40,000 data packets, we found that the query response time is getting more alike. The reason behind that, when we are performing the test on the lower amount of data packet; it might happen that, data has not been uniformly distributed among the cluster. We mentioned earlier that, FL nodes are not so enriched of resources like the cloud. So, for all the FLs, it is going to be a difficult situation to store high volume data packets for a long time.

Interestingly, the Cloud layer resources may augment this gap. Now, to measure the performance of our processing model, we made the query from the end-users' FN node. Initially, the searching of the data has been done into the local FL node, if the query processing fails to retrieve data from the local FL node, then the query hit to the other nodes of the cluster and execute the searching process. If the query requested is satisfied, then it should immediately respond back; otherwise, the request must be propagated to the cloud.

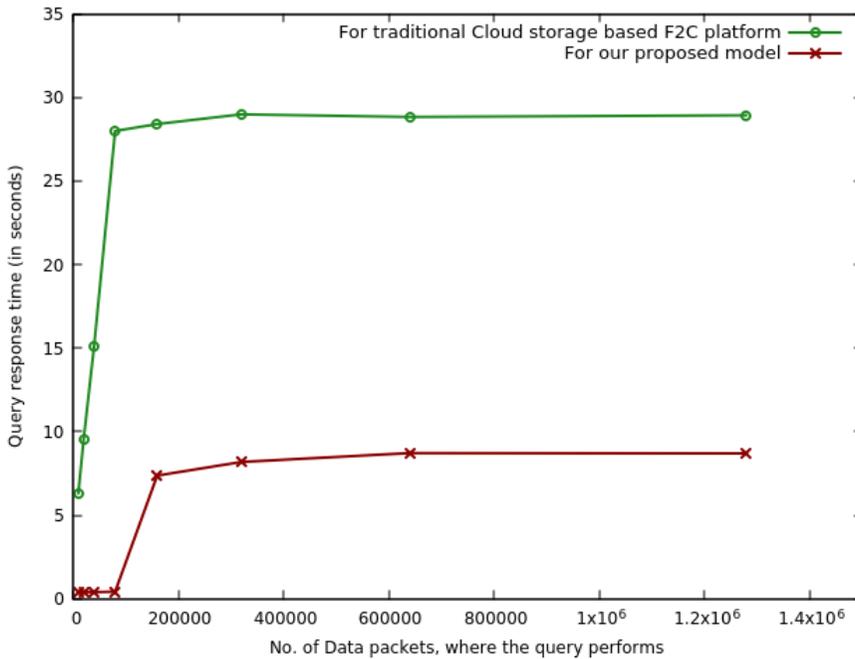


FIGURE 7 Information searching time: Traditional cloud storage-based vs our proposed model.

To compare the performance-efficiency of our proposed model, we also perform the test on a different number of data packets in the traditional cloud-storage based system. In Fig. 7, the red line shows the searching performance of the traditional cloud-storage based framework and the blue line shows the response time for executing the resource information searching process in our proposed model. For the cloud-storage based system, primarily when we performed the searching test on ten thousand distinct data packets, we got a response after 6.309 seconds. As the number of data packets increases, the response time is also increased. Most importantly, by comparing our proposed

model and the traditional cloud-storage based framework, we found that after a certain amount of data packets (more than 80,000) the response time value becomes more constant for both cases. As all the FL nodes reside near to the edge of the network, we observed that our proposing model is capable of providing almost real-time response. Therefore, implementing the distributed database cluster over all the FLs should help the existing F2C framework to be more agile and efficient.

7 | CONCLUSIONS AND FUTURE WORK

This work is presented as an initial footstep, to determine the process for efficiently managing the resource information of the F2C paradigm; so that may explicitly lead to design a comprehensive and proper resource management strategy in that platform. For this reason, we have proposed an introductory version of distributed database-based F2C framework, which uses to store the information that comprehensively characterizes the system resources and service-task. By performing some simulation test, we have measured the performance of our proposed architectural framework. Also, we identify that cost is one of the essential aspects, which has a massive impact on defining the efficient resource management strategy in the F2C paradigm. As earlier said, the validation of the resource matching mechanism and resource allocation strategy is the out of the scope of this work. So in the future work, we are going to focus on for defining a generalized cost model for the F2C paradigm, so that we can build more sophisticated resource allocation and resource management strategy in the F2C platform. Also, in this work, we evaluate the performance of our proposed model by comparing it with the traditional cloud-based execution model. However, to justify the effectiveness of our proposed model, it is necessary to implement our proposed model in real case scenarios (i.e., e-Health application, traffic management system) and execute some tests. For that reason, some additional work is essential. Following this issue and also considering many other challenges constitutes the core of our future work.

references

- [1] Masip-Bruin X, Marín-Tordera E, Tashakor G, Jukan A, Ren GJ. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications* 2016;23(5):120–128.
- [2] Wang M, Perera C, Jayaraman PP, Zhang M, Strazdins P, Shyamsundar R, et al. City data fusion: Sensor data fusion in the internet of things. *International Journal of Distributed Systems and Technologies (IJDST)* 2016;7(1):15–36.
- [3] Sinaeepourfard A, Garcia J, Masip-Bruin X, Marín-Tordera E. Fog-to-Cloud (F2C) Data Management for Smart Cities. In: *Proceedings of 2017 Future Technologies Conference (FTC): 29-30 November 2017, Vancouver, Canada The Science and Information (SAI) Organization*; 2017. p. 162–172.
- [4] Masip-Bruin X, Marín-Tordera E, Juan-Ferrer A, Queralt A, Jukan A, Garcia J, et al. mF2C: towards a coordinated management of the IoT-fog-cloud continuum. In: *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects ACM*; 2018. p. 8.
- [5] Gu L, Zeng D, Guo S, Barnawi A, Xiang Y. Cost efficient resource management in fog computing supported medical cyber-physical system. *IEEE Transactions on Emerging Topics in Computing* 2017;5(1):108–119.
- [6] Cao Z, Lin J, Wan C, Song Y, Zhang Y, Wang X. Optimal cloud computing resource allocation for demand side management in smart grid. *IEEE Transactions on Smart Grid* 2017;8(4):1943–1955.
- [7] Perera C, Qin Y, Estrella JC, Reiff-Marganiec S, Vasilakos AV. Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)* 2017;50(3):32.

- [8] Ni L, Zhang J, Jiang C, Yan C, Yu K. Resource allocation strategy in fog computing based on priced timed petri nets. *IEEE Internet of Things Journal* 2017;4(5):1216–1228.
- [9] Aazam M, Huh EN. Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In: *Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on IEEE*; 2015. p. 687–694.
- [10] Tutschku K, Mehri VA, Carlsson A, Chivukula KV, Christenson J. On resource description capabilities of on-board tools for resource management in cloud networking and NFV infrastructures. In: *2016 IEEE International Conference on Communications Workshops (ICC) IEEE*; 2016. p. 442–447.
- [11] Vasile MA, Pop F, Tutueanu RI, Cristea V, Kołodziej J. Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing. *Future Generation Computer Systems* 2015;51:61–71.
- [12] Manvi SS, Shyam GK. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications* 2014;41:424–440.
- [13] Hahm O, Baccelli E, Petersen H, Tsiftes N. Operating systems for low-end devices in the internet of things: a survey. *IEEE Internet of Things Journal* 2016;3(5):720–734.
- [14] Sanchez L, Muñoz L, Galache JA, Sotres P, Santana JR, Gutierrez V, et al. SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks* 2014;61:217–238.
- [15] Alkhanak EN, Lee SP, Rezaei R, Parizi RM. Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. *Journal of Systems and Software* 2016;113:1–26.
- [16] Yu L, Shen H, Sapra K, Ye L, Cai Z. CoRE: Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost. *IEEE Transactions on Parallel and Distributed Systems* 2017;28(2).
- [17] Rahmanian AA, Dastghaibifard G, Tahayori H. Penalty-aware and cost-efficient resource management in cloud data centers. *International Journal of Communication Systems* 2017;30(8):e3179.
- [18] Mourlin F, Mahmoudi C. Monitoring Architecture for Fog and Mobile Cloud. In: *2018 17th International Symposium on Parallel and Distributed Computing (ISPDC) IEEE*; 2018. p. 109–117.
- [19] Pourghebleh B, Jafari Navimipour N. Towards efficient data collection mechanisms in the vehicular ad hoc networks. *International Journal of Communication Systems* 2019;p. e3893.
- [20] Sengupta S, Garcia J, Masip-Bruin X. Essentiality of resource and service-task characterization in the coordinated fog-to-cloud paradigm. In: *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT) IEEE*; 2018. p. 249–254.
- [21] Zhang H, Zhang Y, Gu Y, Niyato D, Han Z. A hierarchical game framework for resource management in fog computing. *IEEE Communications Magazine* 2017;55(8):52–57.
- [22] Cassandra A. Apache cassandra. Website Available online at <http://planetcassandra.org/what-is-apache-cassandra> 2014;p. 13.

Author Biography



Souvik Sengupta is currently a PhD student in the Department of Computer Architecture (DAC) at the *Universitat Politècnica de Catalunya · UPC BarcelonaTech*. Currently, his research is focus on to identifying the Resource Management strategy in F2C computing. He did his Bachelor Degree (B.Tech) in Computer Science and Engineering from WBUT, India, and followed by the Master's (M.Tech) in Distributed and Mobile Computing from Jadavpur University, India. His research interests are Cloud Computing, the Internet of Things (IoT), Smart Cities, Fog Computing, Big Data, and Resource Management Strategies.



Dr. Jordi Garcia is an Associate Professor in the Department of Computer Architecture at the *Universitat Politècnica de Catalunya · UPC BarcelonaTech*. He has a Master Degree and a PhD Degree, both in Computer Science from the UPC BarcelonaTech. From 2001 to 2004 he was the vice-dean of university extension at the Barcelona School of Informatics (FIB), from 2004 to 2010 he was the vice-dean, head of studies, also at the FIB, and from 2010 to 2013 he was the academic director of the Center of Cooperation for Development. From 2015 to 2016 he has been the Rector's Delegate at the Tongji University SINO Spanish Campus, in China. His current research interests are cloud computing management, new management strategies for smart scenarios and optimization of Big Data processing, parallel systems, optimization techniques for supercomputing, and the optimization of run-time environments.



Dr. Xavi Masip-Bruin got a MSc and Ph.D. degrees in telecommunications engineering both from the Technical University of Catalonia. He is currently an associate professor in the Computer Science department at the UPC, and was invited professor at University Paris-Est Créteil (UPEC) in 2011 and 2012. From 2011 Xavi is the coordinator of the Computer Architecture PhD program at the UPC. He is the founder member of Advanced Network Architectures Lab (CRAAX). His publications include more than 120 papers in national and international refereed journals and conferences. Xavi has given several invited talks (IBM, Huawei, Cisco, UPEC, AdHocNets2013, etc.). Xavi is actively working in the areas of cloud computing, Intelligent Transportation Systems, IoT, Future Internet Architectures and novel network paradigms, network management, data quality as well as in the application of ICT to smart caring (e-health) and animal welfare. More specifically the main focus of his activities is on developing new management strategies for smart scenarios (ITS, cities and health), leading the development of the IVOS platform (<http://www.ivos.cat>).