Data-Centric IoT Services Provisioning in Fog-Cloud Computing Systems

Amir Taherkordi and Frank Eliassen
Department of Informatics
University of Oslo, Norway
{amirhost,frank}@ifi.uio.no

ABSTRACT

Fog computing is mainly proposed for IoT applications that are geospatially distributed, large-scale, and latency sensitive. This poses new research challenges in real-time and scalable provisioning of IoT services distributed across Fog-Cloud computing platforms. Data-centric IoT services, as a dominant type of IoT services in large-scale deployments, require design solutions to speed up data processing and notification, and scale up with the data volume. In this paper, we propose a service-oriented design architecture which is particularly focused on provisioning and processing data-centric IoT services over Fog-Cloud systems. In the proposed architecture, data-centric IoT services are organized in a service integrating tree structure, adhering to the hierarchical fog-based IoT computing models. A service node in the tree is empowered with features for real-time service data notification, local data processing and multi-level IoT data access. The initial results show that, along the design advantages of the proposed model, it does not impose any additional overhead as compared to state-of-the-art solutions.

CCS CONCEPTS

•**Networks** →*Cloud computing;* •**Software and its engineering** →*Data flow architectures; Abstraction, modeling and modularity;*

KEYWORDS

Internet of Things, Fog Computing, Data-Centric Services

ACM Reference format:
Amir Taherkordi and Frank Eliassen. 2017. Data-Centric IoT Services Provisioning in Fog-Cloud Computing Systems. In *Proceedings of The 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA, USA, April 2017 (IoTDI 2017),* 2 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We witness today the convergence of the Internet of Things (IoT) and the Cloud, largely motivated by the need of IoT applications to leverage the scalability, performance, global service accessibility, and pay-as-you-go capabilities of clouds. This has become recently even more complicated with the emergence of fog computing platforms, providing elastic resources and services at the edge of the network [2]. Fog computing is mainly proposed for IoT applications that are geospatially distributed, large-scale, and latency-sensitive, while services such as latency-tolerant and large-scope aggregation can still be efficiently executed in the Cloud.

Research Challenges. Being proposed for large-scale applications with massive number of services and real-time requirements, the way to model and design IoT services at device-, fog-, and cloud-levels is a top priority design challenge [3]. This becomes of particular importance in the case of data-centric services which are characterized by scalable access to services at different network levels, real-time service data notification, and context-aware service provisioning. Three research challenges arise in provisioning of such services. First, with the presence of millions of smart devices (e.g., in smart cities), a challenge is how to model and expose data-centric IoT services in the Fog and the Cloud, process them, and provide their data to interested parties. Second, the latency-sensitivity of fog-based services requires a mechanism for real-time provisioning of service data. For example, in a disaster recovery system, instant changes in local environmental conditions should be propagated to interested applications without any delay. Third, fog devices introduce new location context dimensions to IoT services, such as local processing of IoT data generated in a city district. This raises the challenge of how to develop data-centric IoT services in which location context is a first-class design element.

Related work. The existing work in the area of modeling and provisioning IoT services has either not concretely addressed data-centric services [6], or adopted a uniform and flat view to IoT services and their data, making it difficult to achieve the above goal [8]. Among non-flat models, semantic IoT services are mainly focused on semantic processing of data sets [7] and their relations [5]. In addition, IoT services with real-time requirements are often implemented over publish/subscribe middleware frameworks, focusing only on the type or content of IoT data [1].

Proposed Approach. In this paper, we propose a service-oriented design architecture, which specifically addresses scalable design of data-centric IoT services deployed over Fog-Cloud computing platforms. The essence of the proposed architecture is structuring the description of data-centric IoT services in a hierarchical model (based on the Fog-Cloud network architecture), called Services Hierarchized Over Fogs (SHOF), containing references to services and their real-time and historical data. Each node of SHOF represents a service delivery, processing and notification point for its own children, accessible by other nodes and third-party applications. SHOF is described based on location context attributes of the target IoT application, e.g., physical location of services. Using the proposed architecture, the massive and growing number of IoT services with diverse data types are structured in a hierarchical topology in accordance with the target Fog-Cloud network architecture. This

IoTDI 2017, Pittsburgh, PA, USA
2017. 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

promises scalability by providing different levels of abstractions in providing IoT services and their real-time data.

2 CONCEPTS AND DESIGN MODEL

Based on the typical hierarchical computing model of the Fog and its design objectives, we identify the following key requirements to the design of an architecture for data-centric IoT services provisioning: i) *real-time notification*: a scalable event-based model for interaction in order to filter events at different network levels, from IoT devices to fog nodes and the Cloud, ii) *scalable service data processing*: a context-based IoT data processing model adhering to the Fog network architecture and enabling Fog-level service processing and provisioning, and iii) *historical service data access*: keeping track of historical service data can be highly beneficial when, e.g., the mean value of data is needed or the target service is out of access.

Our design goal is to propose a scalable hierarchical IoT service provisioning model. Figure 1 depicts the general structure of a Fog-Cloud system on the left side and the SHOF service access model on the right side, including its core design elements. Each node of the SHOF is deployed in the Cloud or on a fog device in the target hierarchical Fog network, in order to optimize the scalability and performance of the provided services. We name these types of nodes Fog-Service (FS) nodes. One fog device may contain one or more FS nodes. At the level of leaf nodes, in a hierarchal path, IoT resources are located, called Resource Nodes. Each node of the SHOF maintains a history of data collected in that particular fog boundary of the application, e.g., smoke level of a street during past *n* days. Beyond that, each node can publish real-time data acquired by the node itself and its boundary (i.e., child nodes in the case of a FS node) as events, e.g., a road accident. For both of these data types (i.e., historical data and real-time data), any other nodes of the SHOF can subscribe to receive the data of interest. For example, the traffic control system of street S_1 may depend on the load of the nearby street S_2, thereby the FS node associated to S_1 can subscribe to traffic events published by the FS node of S_2.

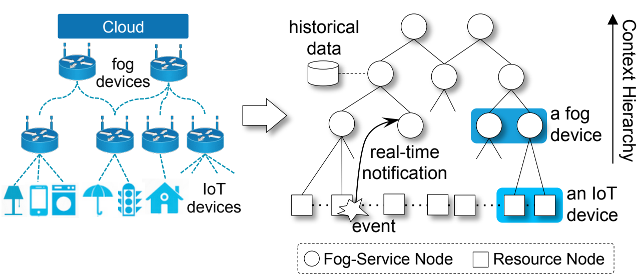

Figure 1: Data-centric IoT services provisioning in SHOF

Each IoT resource has a resource type, which can be sensor, actuator, or tag. The developer needs to build and deploy the SHOF based on the target network's architecture, which represents the location *context hierarchy*. The underlying implementation model of the SHOF is a tree structure whose nodes contain the URI of the corresponding node in the SHOF, along with an API for real-time notification, historical data access, and local data processing. For real-time notifications, two event types are considered: adding a new data item for a service and disconnection of a node (e.g., due to device failure or mobility). Other nodes in the SHOF can subscribe to these event types generated in node FS_i via $FS_i.onAddData()$ and $FS_i.onDisconnect()$, respectively. For historical data access, each node provides an API for time- and date-based access to data, which can be either the data stored under the node itself (i.e., resource nodes) or the data of its child nodes (i.e., FS nodes). The service data processing can use the output of the historical data API for, e.g., calculating the average smoke level of a city district.

3 PRELIMINARY EVALUATION

The realization of the presented approach requires a generic tree structure for populating IoT resources and real-time multilevel access to service data. Firebase [4] is an efficient platform for this purpose—a cloud-based, real-time back-end system that supports various data processing features. Our preliminary evaluation includes comparing the SHOF-based service access model over Firebase with the Orion Context Borker [8], as the state-of-the-art cloud-based IoT service access platform. The evaluation metric is the notification delay when the number of data items added to Orion increases in two data access scenarios: i) original Orion's data access model, and ii) the redesigned SHOF-based model. Figure 2 depicts the behavior of these two scenarios when the number of IoT data entires increases up to 50000. For the smaller number of data items, both scenarios behave similarly, while for bigger numbers we observe a constant difference of 20 ms in notification delay. This experiment shows that, along with the SHOF's other design advantages, its hierarchical model will induce no extra overhead compared to state-of-the-art IoT data processing models.

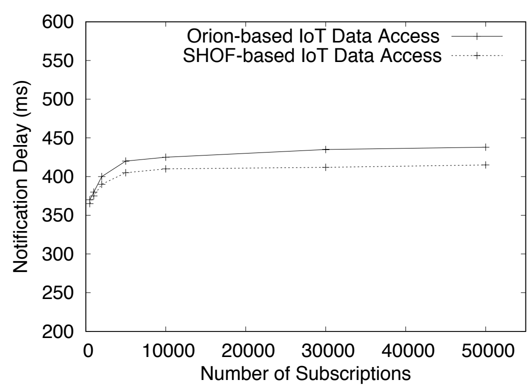

Figure 2: Redesign of Orion's IoT service model to SHOF does not impose any additional notification delay overhead

REFERENCES

[1] K. An et al. A publish/subscribe middleware for dependable and real-time resource monitoring in the cloud. In *SDMCMM Workshop*, 2012.
[2] F. Bonomi et al. Fog computing and its role in the internet of things. In *Workshop on Mobile Cloud Computing*, 2012.
[3] F. Bonomi et al. Fog computing: A platform for internet of things and analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*, 2014.
[4] Firebase Cloud Platform. http://www.firebase.com/.
[5] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool, 1st edition, 2011.
[6] F. Li et al. Efficient and scalable iot service delivery on cloud. In *CLOUD*, 2013.
[7] J. Liu and F. Zhao. Towards semantic services for sensor-rich information systems. In *Broadband Networks*, 2005.
[8] Orion Context Broker. http://fiware-orion.readthedocs.io.