Poster Abstract: Supporting Fog/Edge-based Cognitive Assistance IoT Services for the Visually Impaired

Shashank Shekhar*
Siemens Corporate Technology
Princeton, NJ
shashankshekhar@siemens.com

Ajay Chhokra
Vanderbilt University
Nashville, TN
ajay.d.chhokra@vanderbilt.edu

Hongyang Sun
Vanderbilt University
Nashville, TN
hongyang.sun@vanderbilt.edu

Aniruddha Gokhale
Vanderbilt University
Nashville, TN
a.gokhale@vanderbilt.edu

Abhishek Dubey
Vanderbilt University
Nashville, TN
abhishek.dubey@vanderbilt.edu

Xenofon Koutsokos
Vanderbilt University
Nashville, TN
xenofon.koutsokos@vanderbilt.edu

ABSTRACT

The fog/edge computing paradigm is increasingly being adopted to support a variety of latency-sensitive IoT services, such as cognitive assistance to the visually impaired, due to its ability to assure the latency requirements of these services while continuing to benefit from the elastic properties of cloud computing. However, user mobility in such applications imposes a new set of challenges that must be addressed before such applications can be deployed and benefit the society. This paper presents ongoing work on a dynamic resource management middleware called URMILA that addresses these concerns. URMILA ensures that the service remains available despite user mobility and ensuing wireless connectivity issues by opportunistically leveraging both fog and edge resources in such a way that the latency requirements of the service are met while ensuring longevity of the battery life on the edge devices. We present the design principles of URMILA's capabilities and a real-world cognitive assistance application that we have built and are testing on an emulated but realistic IoT testbed.

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**;
• **Computer systems organization → Cloud computing**;

KEYWORDS

Fog/Edge Resource Management, User Mobility, Latency

ACM Reference Format:
Shashank Shekhar, Ajay Chhokra, Hongyang Sun, Aniruddha Gokhale, Abhishek Dubey, and Xenofon Koutsokos. 2019. Poster Abstract: Supporting Fog/Edge-based Cognitive Assistance IoT Services for the Visually Impaired. In *IoTDI '19: Conference on Internet of Things Design and Implementation, April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3302505.3312592

*Work performed during doctoral studies at Vanderbilt University

IoTDI '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6283-2/19/04.
https://doi.org/10.1145/3302505.3312592

1 INTRODUCTION

Rapid advances in Internet of Things (IoT) technologies have given rise to a variety of new services that have significant societal impact. Consider a real-time object detection application targeted towards visually impaired for cognitive assistance. The associated wearable IoT equipment such as an eyewear enhances their knowledge about the surroundings. Such a service will need to frequently capture images of the surroundings using the wearable equipment, process and analyze these frames, and subsequently provide feedback (e.g., audio and haptics) to the user.

Designing such a service is a hard problem for a number of reasons. For instance, the design must decide where to process the frames. If these frames were to be processed always at the edge (e.g., on a smartphone carried by the user), then the edge resource must typically be configured with a pre-trained image classification model which is needed to identify objects in the captured frames. When compared to powerful server resources in a cloud, edge resources tend to be significantly limited in the amount of memory, computation speed and battery power. Thus, it is unreasonable to expect very large pre-trained models to be stored on edge devices nor can we expect highly compute intensive model execution and image classification tasks on such edge devices and that too performed repeatedly for every frame that is captured as the user moves. Ensuring the longevity of the battery life on the edge device is a key requirement.

An alternative is to always offload the computations onto fog resources, which are a collection of servers that are more powerful and richer in resources relative to edge resources. However, relying only on fog resources is fraught with challenges. First, even if the service chooses a fog resource in the vicinity of the user, which itself needs a discovery step to identify a fog resource, there is no guarantee that that fog resource currently has enough capacity to handle a new task because it may already be heavily utilized by other IoT services. Second, since the user is mobile, the user is very likely to go out of range of the selected fog resource which means another fog resource must be discovered and the service state transferred while ensuring that the service remains available when it is needed. It is possible that during this handoff some frames may not get processed thereby compromising the safety of the user. Finally, both due to fluctuating wireless signals and potentially no

signal range with any fog resource, several captured frames may not get processed at all, which exacerbates the problem even further.

In summary, a fog-only or edge-only solution will not yield the desired service functionality and service-level guarantees. Thus, an approach that can intelligently and dynamically switch between fog and edge resources as the user moves while meeting the service-level guarantees is needed. To that end, we are designing a dynamic and adaptive resource management middleware solution for IoT called *URMILA (Ubiquitous Resource Management for Interference and Latency-Aware services)*. URMILA's dynamic resource management algorithms (a) account for the constraints of the edge resources (e.g., remaining battery charge, available memory and computing power) so that the appropriate service logic can be executed on the edge resources when the service has no choice but to execute on the edge resource, (b) enable discovery and use of those fog resources that are reachable over wireless links and that also have enough available capacity to host the service such that the detrimental effects of performance interference caused due to multiple co-located services [1, 2] is minimized, (c) avoid the need to migrate any application state between fog and edge resources or hand-off state between fog resources as the user moves which otherwise will waste resources and degrade performance.

2 ONGOING WORK

Our ongoing work comprises two thrusts as decribed below.

2.1 Service Design

We are designing a soft real-time object detection, cognitive assistance application targeted towards the visually impaired. Advances in wearable devices and computer vision algorithms have enabled cognitive assistance and augmented reality applications to become a reality, e.g., PivotHead's SeeingAI and Gabriel [3] that leverage Google Glass and the fog devices.

To that end and to showcase some variety in edge devices uses, our first design uses an Android smartphone that inter-operates with a Sony SmartEyeGlass, which is used to capture video frames as the user moves in a region and provides audio feedback after processing the frame. The second design uses a Python application running on Linux-based board devices such as MinnowBoard with a Web camera. In both the implementations, we maintain two different real-time object detection algorithms from Tensorflow for image processing: MobileNet, which is smaller in size but less accurate is executed on the edge device, while Inception V3, which is larger in size but more accurate executes on a fog device. Two models help avoid model and data movement between the edge and fog devices, and overcomes the limitation of edge devices, which constrain storage and execution of large models.

2.2 Middleware Design

Figure 1 shows the model-predictive, control-based, dynamic and adaptive resource management architecture of URMILA. We formulate and solve an optimization problem[1] in the URMILA middleware that trades-off execution of image classification on edge or fog resource for the cognitive assistance service. The decision is based on available remaining battery power on an edge device, availability of a fog resource with minimal interference from co-located applications within signal strength of a user, predicted path taken by the mobile user and the estimated duration of its connectedness to that selected fog device, and communication costs to reach the fog device. Since this optimization problem is NP Hard, URMILA implements a heuristic-based server selection algorithm.

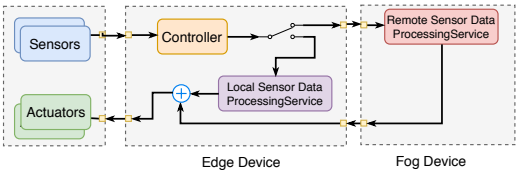

Figure 1: URMILA Methodology for Adaptive Execution of Service Logic

URMILA provides a range of individual solutions to obtain the different inputs needed to solve the optimization problem. These include predicting the path taken by the user, and based on this predicted path, determine the likely fog resources that the user is likely to be reachable over the wireless channel. Accordingly, URMILA comprises a wireless signal strength estimator to estimate the signal strength of a user from a given point on its predicted path to one or more available fog devices on the route. By keeping track of instantaneous loads on each fog device through its monitoring infrastructure [4] that is deployed on each fog device, URMILA also provides a latency estimation algorithm to determine the execution time to execute an image classification task on the available fog resources and round trip latency to the user, which is then used in choosing a fog device with the least performance interference for a given route segment. If no fog device is available, execution will be carried out on the edge device until a next fog device is found. URMILA is available in open source at github.com/doc-vu.

ACKNOWLEDGMENTS

This work is supported in part by NSF US Ignite CNS 1531079 and AFOSR DDDAS FA9550-18-1-0126. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF or AFOSR.

REFERENCES

[1] Christina Delimitrou and Christos Kozyrakis. 2013. Paragon: QoS-aware scheduling for heterogeneous datacenters. In *ACM SIGPLAN Notices*, Vol. 48. ACM, 77–88.

[2] Dejan Novaković, Nedeljko Vasić, Stanko Novaković, Dejan Kostić, and Ricardo Bianchini. 2013. DeepDive: Transparently Identifying and Managing Performance Interference in Virtualized Environments. In *Proceedings of the 2013 USENIX Conference on Annual Technical Conference (USENIX ATC'13)*. USENIX Association, Berkeley, CA, USA, 219–230. http://dl.acm.org/citation.cfm?id=2535461.2535489

[3] Mahadev Satyanarayanan, Zhuo Chen, Kiryong Ha, Wenlu Hu, Wolfgang Richter, and Padmanabhan Pillai. 2014. Cloudlets: at the leading edge of mobile-cloud convergence. In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 1–9.

[4] Shashank Shekhar, Ajay Chhokra, Anirban Bhattacharjee, Guillaume Aupy, and Aniruddha Gokhale. 2017. INDICES: Exploiting Edge Resources for Performance-Aware Cloud-Hosted Services. In *IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*. Madrid, Spain, 75–80. https://doi.org/10.1109/ICFEC.2017.16

[1] Optimization problem formulation and preliminary results not shown due to space constraints.