RUCON LiveLab: Distributed Analytics for Edge Computing

Ivan Lujic, Vincenzo De Maio, Atakan Aral, Fani Basic, Josip Zilic, Ivona Brandic
Institute of Information Systems Engineering, Vienna University of Technology
{first name}@ec.tuwien.ac.at

***Abstract*—The edge computing paradigm has been proposed as a way to address near real-time requirements of typical IoT applications by providing cloud functionalities closer to the source of data. In this work, we describe the testbed that we build for typical edge applications. First, we describe the hardware and software configuration of Rucon LiveLab; then, we describe a possible use case for Rucon LiveLab. Finally, we describe open challenges in distributed analytics for edge computing.**

I. INTRODUCTION

The edge computing paradigm has been proposed to meet the strict latency and accuracy requirements of modern applications by extending cloud functionalities closer to the source of data. Nowadays, it is possible to process data closer to data sources thanks to technological advances that allow placement of computation, network and storage capabilities to edge nodes, i.e., micro data centers or resource-constrained devices such as Raspberry Pis. To cope with the ever-growing application requirements and user needs, in the near future, we can expect many edge-deployed and distributed clusters managed by different providers as in the multi-cloud concept. The new edge computing paradigm should ensure the adaptive placement of data analytics tasks and application instances across different infrastructures to keep overall system performance under control. Edge-deployed clusters can be heterogeneous, e.g., containing different initial capacities and different availability of resources over a certain period of time.

In this work, we describe RUCON LiveLab, the distributed infrastructure that we employ as edge applications testbed. First, we describe the architecture of RUCON LiveLab and the technologies that we use for our analytics. Afterward, we describe a use case for Rucon LiveLab in the IoT context. Finally, we describe future work and open challenges.

II. RUCON LIVELAB

A fog computing testbed for rapid prototyping fog computing components has been described in [1]. The proposed system is called PiFogBed and it is designed for mobile computing. In comparison with the centralized architecture of PiFogBed, we propose a distributed architecture with three sub-clusters, to simulate a more geographically distributed environment. Also, PiFogBed relies also on Cloud nodes and other additional components, like mobile nodes, and targets medical applications. In our work, we focus on more computationally intensive applications, such as video analytics, to test the capability of our system to respect near real-time constraints on applications with higher demands.

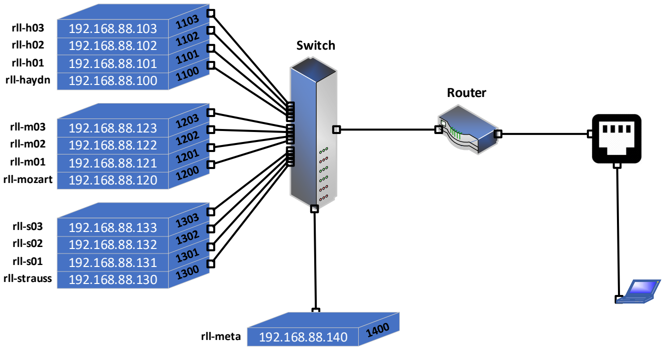

Fig. 1. System Configuration of the RUCON LiveLab.

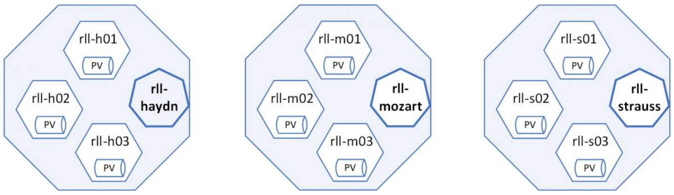

Fig. 2. Kubernetes Level Architecture.

Figure 1 shows the physical architecture on which we plan to simulate novel approaches using 12 Raspberry Pis 3B+ separate into 3 stackable cases, each containing 1 master and 3 worker nodes. Each RPi is equipped with 1 GB RAM memory and a Quad-Core ARM processor running at 1.4 GHz. All RPis are connected to the network with Netgear 24-Port 10-Gigabit Switch and an Ethernet router. Further, we plan to utilize two additional in-house servers (rll-mdc01 and rll-mdc02) equipped with 256 GB RAM memory and 24-core Intel Xeon E5 processor running at 2.2 GHz, to simulate an edge micro data center. Since the Kubernetes cluster is based on master-worker architecture, Figure 2 illustrates the setup in which every cluster consists of 1 master and 3 worker nodes.

Before building the testbed, a set of technologies, tools, and languages was used in edge context to set up a virtual environment for testing purposes, including:

- **Kubernetes** is a platform that is one of the widely used open-source orchestrators that automates deployment and management of multiple containerized applications across multiple machines. Kubernetes can be installed with minikube as a multi-node cluster on the localhost.

- **Docker** is a container platform used to build and isolate the applications and corresponding stack of services on containers, that is, standalone execution environments.
- **Vagrant** represents a tool for managing virtual machine environments. One of the typical providers to set up virtual machines is VirtualBox. Additionally, Ansible playbooks are used in combination with Vagrant to install needed packages and tools (e.g., Kubernetes, Docker). Ansible playbooks, as later Kubernetes deployment manifest files, are written in YAML (Yet Another Markup Language) as it is often used for configuration files.

III. USE CASE SCENARIO

Many IoT applications require fast response times and real-time decisions. However, data often travel a long distance from sensors to a cloud data center for processing, while sending results back to users. In this use case, we aim to design a distributed data analytic framework running on edge nodes such as Raspberry Pis and micro data centers. By integrating these edge resources efficiently into data processing, we can reduce response time and network bandwidth. Finally yet importantly, there is an increasing demand for data analytics that can be dynamically and modularly applied to collected IoT data in real-time. Especially, machine learning algorithms [2] from a shared and reusable toolbox should be made available to users in order to facilitate their data analytics tasks. These tasks include video analytics for surveillance to identify missing/wanted individuals or detect unusual activities by patients.

In addition, the wide-area network will likely suffer from data congestion in the near future due to 20 billion forecasted IoT devices. Due to network congestion or node failures, it becomes important to be aware of failure probabilities [3], especially in emerging edge computing. Thus, instead of traditionally performing centralized data analytics, the edge intelligence [4] should strive for dynamic placement of data analytics tasks across different nodes at runtime. The framework should enable parallelization for data analytics, allowing subtasks to be processed closer to the data sources and thus reducing response time and costly data transfer to the cloud.

Our main objectives through this testbed are:

- building a novel distributed edge analytics framework;
- allowing dynamic and self-adaptive placement of processing components across edge nodes.

Distribution of data analytics depends on four aspects, namely, (i) workload size in which data are constantly produced; (ii) time-sensitivity or urgency level to deliver data analytics results; (iii) resource availability; and (iv) complexity of data analytics tasks. However, each scenario has different requirements and challenges for performing data analytics:

- Which algorithms to apply and how to configure different input parameters?
- How to preprocess, filter data, and which data to use?
- How to select the right amount and type of resources based on complexity and runtime demands?

A typical example of applications that may benefit from Edge analytics are video-surveillance and driving assistance since these applications require video streaming analytics tasks (e.g., object recognition, to prevent collisions and accidents or to identify suspects in an area). Object recognition tasks have to be performed with strict latency requirements, in order to avoid disasters, especially in the case of driving assistance and collision detection. For these reasons, these tasks cannot afford the latency caused by the round trip between streaming devices and the remote Cloud. Pushing intelligence to the Edge, closer to these devices, will significantly reduce latency and allow timely reactions [5]. Our group, Magenta and Swarco are implementing an intelligent traffic safety solution based on 5G, where the surroundings are recorded by cameras on the traffic lights and relevant events are reported to the vehicles. Data processing takes place locally within the traffic light system to protect the privacy of road users.

IV. OUTLOOK

Currently, we are extending RUCON LiveLab to collect samples from different devices and different user behaviors. The computational facilities of the RUCON LiveLab will include a computational backend, 45 edge nodes and IoT devices that are spread in the computer science building. We aim to collect real data traces by utilizing real human environments. We will perform various code offloading strategies including machine learning applications such as face recognition or navigator application as described in our recent ICPE paper [6]. Further, we plan to use RLL for evaluating proposed edge data management strategies [7] in the context of efficient predictive analytics for critical and proactive IoT systems.

ACKNOWLEDGMENT

This work has been partially funded through the RUCON project (Runtime Control in Multi Clouds), FWF Y 904 START-Programm 2015, European Union Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No.83894, and two netidee scholarships by the Internet Foundation Austria.

REFERENCES

[1] Q. Xu and J. Zhang, "piFogBed: A fog computing testbed based on Raspberry Pi," in *IEEE International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2019, pp. 1–8.

[2] R. Mayer and H.-A. Jacobsen, "Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–37, 2020.

[3] A. Aral and I. Brandic, "Dependency mining for service resilience at the edge," in *IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 228–242.

[4] Z. Zhou *et al.*, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[5] C. Zhu *et al.*, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1–9.

[6] V. De Maio and I. Brandic, "Multi-objective mobile edge provisioning in small cell clouds," in *ACM/SPEC International Conference on Performance Engineering (ICPE)*, 2019, pp. 127–138.

[7] I. Lujic, V. De Maio, and I. Brandic, "Resilient edge data management framework," *IEEE Transactions on Services Computing*, 2019.