A Demo of Workload Offloading in Mobile Edge Computing using the Reliable Server Pooling Framework

Thomas Dreibholz*, Somnath Mazumdar†
* SimulaMet – Simula Metropolitan Center for Digital Engineering
Center for Resilient Networks and Applications
Pilestredet 52, 0167 Oslo, Norway
† Department of Digitalization
Copenhagen Business School
Solbjerg Plads 3, 2000 Frederiksberg, Denmark
Email: dreibh@simula.no, sma.digi@cbs.dk

***Abstract*—Mobile Edge Computing (MEC) places cloud resources nearby the user, to provide support for latency-sensitive applications. Offloading workload from resource-constrained mobile devices (such as smartphones) into the cloud ecosystem is becoming increasingly popular. In this demonstration, we show how to deploy a mobile network with OPENAIRINTERFACE and OPEN SOURCE MANO. We also demonstrate how to adapt the Reliable Server Pooling (RSerPool) framework to efficiently manage MEC as well as multi-cloud resources.**[1 2 3]

***Index Terms*—Cloud, Mobile Edge Computing, NVF, Open Source MANO, OpenAirInterface, Reliable Server Pooling (RSerPool)**

I. Introduction

For executing a wide range of user applications, Cloud computing platforms are widely accepted due to their scalability, flexibility as well as cheaper usage prices. However, the Cloud platform is not ideal for latency-sensitive applications. Furthermore, for higher performance, the requirements of user applications are also changing. Thanks to the evolution of mobile technology (such as hardware and cellular networks), running applications on mobile devices – particularly on smartphones – are becoming a new norm. Mobile Edge Computing (MEC) [1] is a way to solve the latency issues, by placing cloud resources nearby a connected user device. This leads to the question of how to realise services for MEC. However, such mobile devices host *limited* computational resources, i.e. processor, primary memory and storage space. Nowadays, the graphical user interface (GUI) of a cloud service is a popular candidate application-type running on smartphones. The challenge of such cases is that the latency between user devices and the cloud becomes relevant. If the latency is too high, the reaction of the application to user interaction gets inconveniently slow.

Reliable Server Pooling (RSerPool) [2] is an IETF standard for a lightweight server pooling approach. In this demo work, we have customised the existing RSerPool approach to tackle the problem of organising MEC services. In [3], we presented a proof-of-concept for realising RSerPool-based workload offloading with MEC. In this proposed demo (see details in Section IV), we simulate a realistic use case scenario using RSerPool and also visually demonstrate the intuitiveness of such an approach. We are going to present this idea to the audience by using an interactive live demo.

[1]This project has received partial funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 815279.

[2]Parts of this work have been partially funded by the Research Council of Norway (Forskingsrådet), prosjektnummer 208798/F50.

[3]The authors would like to thank Ann Edith Wulff Armitstead for her friendly support.

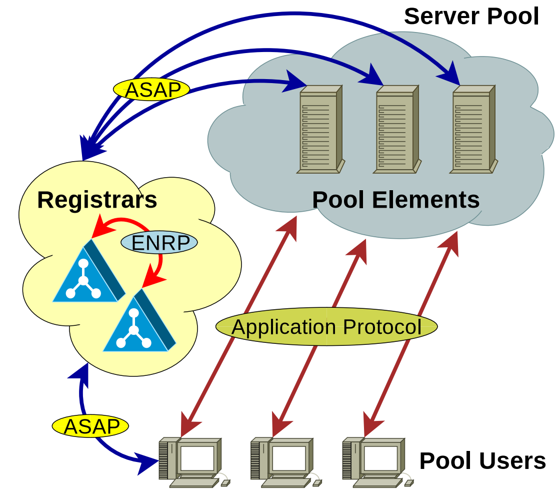

Fig. 1. The representation of RSerPool architecture

II. Background

A. Reliable Server Pooling (RSerPool)

Reliable Server Pooling (RSerPool) [4]–[6] is a generic, application-independent framework for server redundancy and session handling. A particular property of RSerPool is to be simple and lightweight, making it suitable for devices with very limited hardware resources. RSPLIB[4] [6] is an open-source implementation of RSerPool.

The RSerPool architecture [2] shown in Figure 1, where a number of servers, each providing a certain service, form a pool. Servers of a pool are denoted as Pool Elements (PE). Within its operation scope, a pool is identified by its unique

[4]RSPLIB: https://www.uni-due.de/~be0001/rserpool/.

Pool Handle (PH, e.g. a string like "Application XY"). The set of all pools within an operation scope, the so-called handlespace, is managed by Pool Registrars (PR, shortly denoted as registrars). The PRs synchronise the handlespace by using the Endpoint haNdlespace Redundancy Protocol (ENRP) [6]. Servers can dynamically register to, and deregister from, a pool at any PR of their operation scope, by using the Aggregate Server Access Protocol (ASAP) [6].

Clients, which are denoted as Pool Users (PU), use ASAP to access the resources of a pool. A PR can be asked for the selection of PE(s) in its operation scope. This selection is performed by using a pool-specific pool member selection policy [4], [6], [7]. Furhermore, ASAP can be used for realising a Session Layer functionality between a PU and a pool. In this case, ASAP may also support the actual Application Layer protocol with state synchronisation [4], [6].

B. Open Source MANO and the SimulaMet OAI EPC

Network Function Virtualisation (NFV) [8] is a crucial part of 5G networks: Network functionalities can be realised as Network Services (NS), which are composed of Virtual Network Functions (VNF). NSs can then be instantiated as Virtual Machines (VM) in data centres. This allows for very high flexibility: NSs can be instantiated dynamically when needed and removed when not needed anymore. Furthermore, VNF instances can be scaled as needed. OPEN SOURCE MANO[5] (OSM) [9] is a framework for managing and orchestrating NFV. It is the orchestration platform from ETSI. It utilises an underlying Network Function Virtualisation Infrastructure (NFVI) for instantiating the Virtual Deployment Units (VDU) as VMs. A commonly used NVFI is OPENSTACK, but OSM supports other frameworks as well.

Based on OSM, we developed a VNF for the Enhanced Packet Core (EPC) of OPENAIRINTERFACE[6] (OAI), denoted as SIMULAMET OAI VNF[7] [10]. In particular, it can be used to easily realise a tailor-made EPC for custom network testbed setups. The configuration flexibility of our VNF can be utilised easily by using it in new NSs (e.g. by adding MEC resources). We will explain the details next in Section III.

III. PROPOSED APPROCH: MEC WITH RSERPOOL

Here, we are going to explain our approach from [3] to handle the workload offloading challenge. The basic scenario is illustrated in Figure 2: the UEs run a certain application, which is demanding when it comes to computation and/or storage. They have very limited computing resources. MEC resources to support this application are available nearby the user. Furthermore, it may also be possible to have additional resources in public (multi-)clouds [11], [12] somewhere in the Internet.

Our approach to applying RSerPool is illustrated in Figure 3. The resources are added into a pool (identified by its PH) "Fractal Generator Pool". This means, the servers in the MEC, and (possibly) also auxiliary servers in a public (multi-)cloud, become PEs of this pool. From the implementation perspective, it is trivial also to start a server instance on the UE

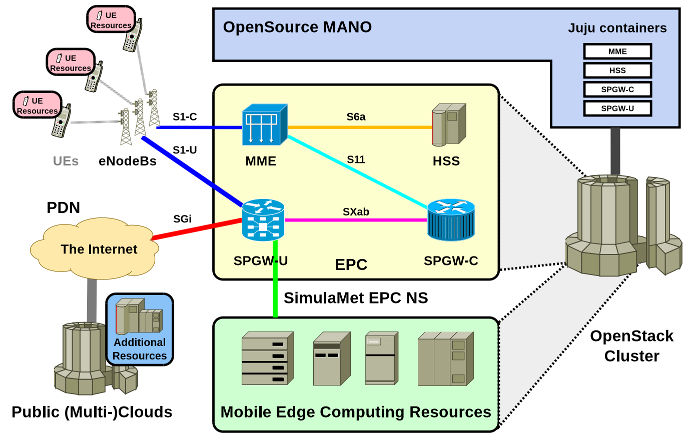

Fig. 2. Testbed Setup with EPC, OPEN SOURCE MANO and Cloud

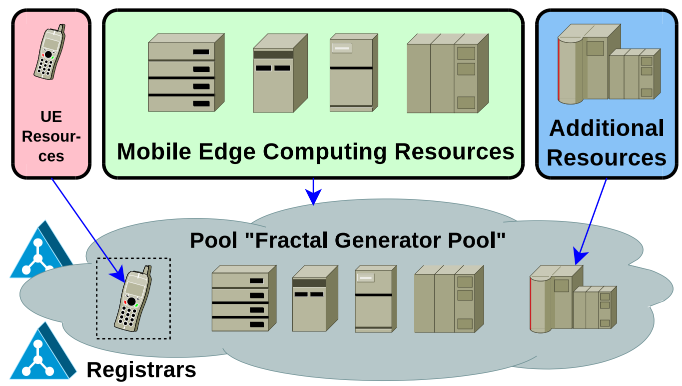

Fig. 3. RSerPool with Mobile Edge Computing and public multi-Cloud

itself if the application allows this, due to resource constraints on the UE. RSerPool itself is very lightweight, which means it will not add a large management overhead [6]. The PE on the UE would then be part of the pool, too. If everything else fails, for instance, due to loss of network coverage, it would allow running the application on the UE device, albeit with reduced performance. The PU-side of the application then just needs to pick a *suitable* PE from the pool (or even use multiple PEs in parallel) to use their service. The choice of a suitable PE is, as described in Subsection II-A and [3], the goal of a suitable pool policy to handle the differences between UE PE, MEC PEs as well as public (multi-)cloud PEs.

Finally, PRs are needed to manage the handlespace. At least one PR needs to run within the core network (e.g. as part of the MEC setup). Another PR can run locally to the additional PEs in the public (multi-)cloud. To allow the UE to run a local PE without network coverage, the UE would need to run its own PR instance. It is lightweight (i.e. having only low memory and CPU requirements).

IV. DEMONSTRATION SCENARIO AND SETUP

An overview of our proposed demo is presented in Figure 4. The demo consists of:

- OPENAIRINTERFACE-based EPC, instantiated by the SIMULAMET OAI VNF [3], [10] in a local OPENSTACK

[5] OPEN SOURCE MANO: https://osm.etsi.org.
[6] OPENAIRINTERFACE: https://www.openairinterface.org.
[7] SIMULAMET OAI VNF: https://github.com/simula/5gvinni-oai-ns.

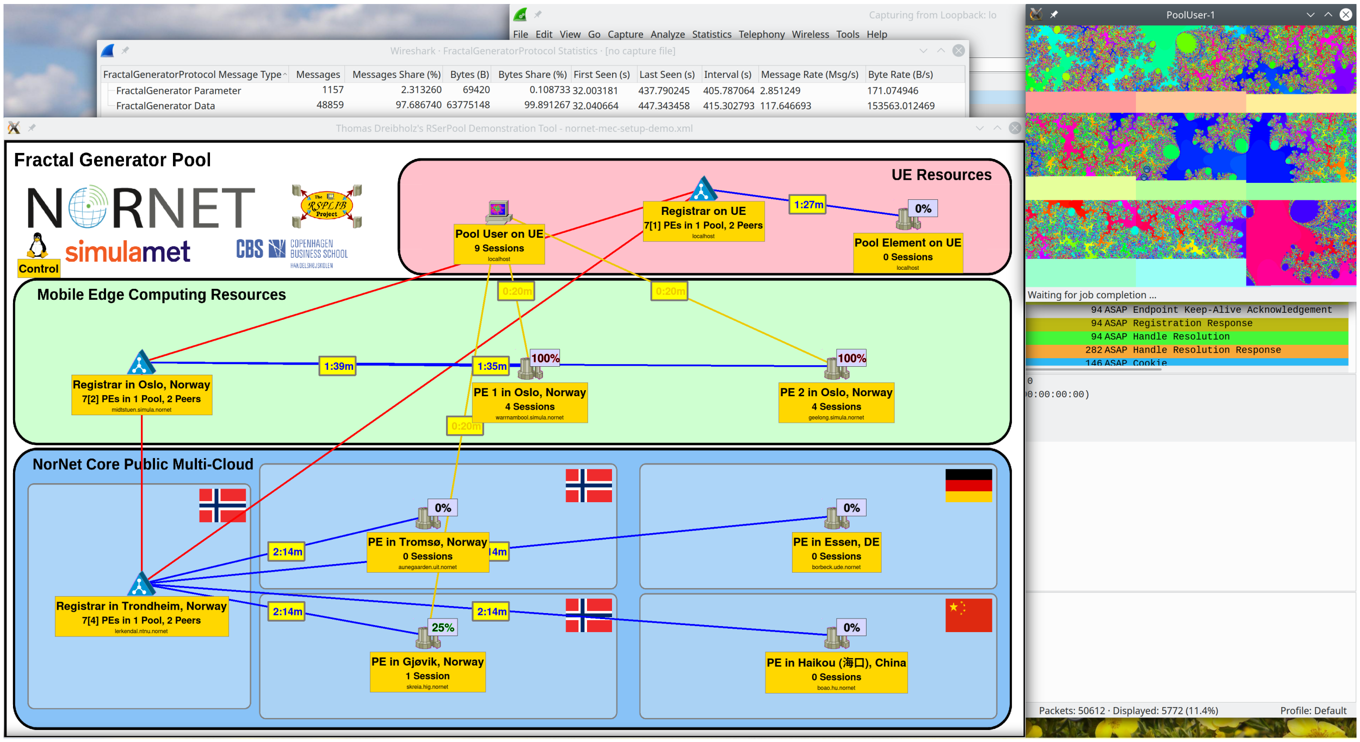

Fig. 4. The User Interface of our Interactive Live Demo

cluster, orchestrated by OSM;

- OPENAIRINTERFACE-based eNodeB with Software-Defined Radio (SDR) hardware;
- UE (laptop) with an interactive user interface as shown in Figure 4, connecting to the eNodeB and EPC via LTE;
- UE resources (PE on UE) in our lab (in Oslo, Norway);
- Local MEC resources: 2 PEs (in Oslo, Norway);
- Public Multi-Cloud resources: 4 PEs (in Tromsø and Gjøvik, Norway; Essen, Germany; and Haikou, China).

The client application on the UE requests the computation of fractal graphics, which are computed by PEs in the pool. The user interface illustrates the connections established in the setup. Further details can be provided by interactive WIRESHARK [13] traces. We also extended WIRESHARK by dissectors for the used application protocols as well as by advanced statistics features for the used protocols.

V. CONCLUSIONS

In our proposed demo, we would like to showcase how to set up and manage our own 4G mobile broadband setup with RSerPool-based MEC and public multi-cloud resources for an interactive fractal graphics computation application.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, 2017.

[2] P. Lei, L. Ong, M. Tüxen, and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols," IETF, Informational RFC 5351, 2008.

[3] T. Dreibholz and S. Mazumdar, "Reliable Server Pooling Based Workload Offloading with Mobile Edge Computing: A Proof-of-Concept," in *Proceedings of the 2nd International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC) in conjunction with the 35th International Conference on Advanced Information Networking and Applications (AINA)*, vol. 3, Toronto, Ontario/Canada, 2021.

[4] T. Dreibholz and E. P. Rathgeb, "Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework," *International Journal on Advances in Internet Technology (IJAIT)*, vol. 2, no. 1, 2009.

[5] T. Dreibholz, X. Zhou, M. Becke, J. Pulinthanath, E. P. Rathgeb, and W. Du, "On the Security of Reliable Server Pooling Systems," *International Journal on Intelligent Information and Database Systems (IJIIDS)*, vol. 4, no. 6, 2010.

[6] T. Dreibholz, "Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture," Ph.D. dissertation, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, 2007.

[7] T. Dreibholz and M. Tüxen, "Reliable Server Pooling Policies," IETF, RFC 5356, 2008.

[8] M. Veeraraghavan, T. Sato, M. Buchanan, R. Rahimi, S. Okamoto, and N. Yamanaka, "Network function virtualization: A survey," *IEICE Transactions on Communications*, vol. E100-B, 2017.

[9] A. Reid, A. González, A. E. Armengol, G. G. de Blas, M. Xie, P. Grønsund, P. Willis, P. Eardley, and F.-J. R. Salguero, "OSM Scope, Functionality, Operation and Integration Guidelines," ETSI, White Paper, 2019.

[10] T. Dreibholz, "Flexible 4G/5G Testbed Setup for Mobile Edge Computing using OpenAirInterface and Open Source MANO," in *Proceedings of the 2nd International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC) in conjunction with the 34th International Conference on Advanced Information Networking and Applications (AINA)*, Caserta, Campania/Italy, 2020.

[11] T. Dreibholz, S. Mazumdar, F. Zahid, A. Taherkordi, and E. G. Gran, "Mobile Edge as Part of the Multi-Cloud Ecosystem: A Performance Study," in *Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Pavia, Lombardia/Italy, 2019.

[12] J. Hong, T. Dreibholz, J. A. Schenkel, and J. A. Hu, "An Overview of Multi-Cloud Computing," in *Proceedings of the 1st International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC) in conjunction with the 33rd International Conference on Advanced Information Networking and Applications (AINA)*, Matsue, Shimane/Japan, 2019.

[13] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress, 2007.