Poster: EPS – Edge-hosted Personal Services for Mobile Users

Pengzhan Hao, Yongshu Bai, Xin Zhang, and Yifan Zhang
Department of Computer Science
SUNY Binghamton
Binghamton, NY
{phao3, ybai4, xzhang99, zhangy}@binghamton.edu

Introduction. Being powered by battery and the reliance on expensive cellular data while users are on the move are holding *smart mobile devices* from being widely used in those long-lasted, computation-intensive, or highly network-reliant usage scenarios. In the meantime, more and more mobile workloads and optimizations now rely on the cloud, such as mobile cloud offloading, cloud-based mobile web optimization, and cloud-based network traffic redundancy reduction. However, it is hard to perform *large-scale* and *personalized* support or optimization for mobile workloads without significant computation resource increase on the cloud, as well as higher bandwidth requirement on the networks.

We target solving the above two problems by enabling *edge-hosed personalized services* (EPS for short). The idea of EPS is twofold. One is enabling developer-customizable and power-and-traffic-efficient network communication on the "last-hop" communication between mobile devices and the network edge. The other is distributing cloud services for mobile workloads to network edge, so that they can be done on a personalized basis, while enjoying much lower communication latency to/from mobile devices. Our approach is to deploy on the network edge certain computation endpoints (named as "service carriers"), each of which performs some specific service for individual users. The goal is to build a practical, secure, efficient and scalable framework that supports *customized communication protocols* between network edge and mobile devices, as well as *customizable collaborative workloads/optimizations* on mobile devices and the edge.

Motivation usage scenarios. Many mobile workloads can be significantly benefited from customized communication protocols for reduced network traffic and prolonged battery life. One such scenario is *cloud-storage-backed mobile document editing*: a user edits office suite documents on his smartphones/tablets. The content changes made by the user are sent to cloud storage, and can be further synchronized in real-time to other devices owned by the user or his collaborators. We find that all current cloud storage services transmit almost the entire file even for one character change, despite the fact that deduplication and delta-encoding are being applied. An effective solution would require changes on the cloud storage providers side, which can take a long time to be deployed. With an EPS that enables customized communication protocol for such scenario, we can significantly reduce network traffic generated by mobile devices. There are many other usage scenarios that can be benefited from EPS. For example, EPS can enable personalized web data reduction (e.g., through compression, image transcoding), personalized web caching on the edge, and personalized preconnecting and prefetching for mobile web.

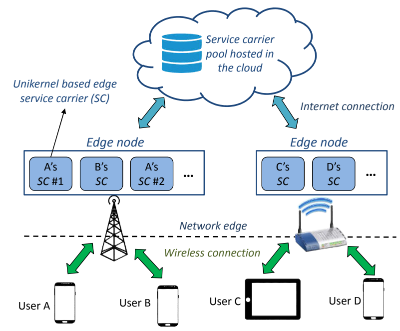

Figure 1: EPS Architecture

EPS architecture. Figure 1 shows the architecture of EPS. When users move to a new location, their service carriers can be migrated from the previous edge node to the current one through the cloud-based service carrier pool. The service carrier is designed to be a Unikernel [1]. We choose Unikernel because of two main reasons. First, Unikernels are extremely small and lightweight, so that they can be migrated to and booted on resource-constrained edge nodes quickly. Second, Unikernels provide perfect resource isolation. This ideal to ensure user data security and privacy on an edge node, which also runs other users' services.

Ongoing work. We have implemented an EPS for the cloud-storage-backed mobile document editing usage scenario mentioned previously. Our current implementation of the EPS's service carrier is a Unikernel running on Xen hypervisor on an x86 desktop server. We are working on porting our implementation onto ARM boards. In the meantime, we have also started the process of designing and building a programming and system framework that allows mobile application developers to take advantage of EPS to improve mobile user experiences.

Acknowledgments. This work is being supported in part by NSF Award #1566375.

References

[1] A. Madhavapeddy and D. J. Scott. Unikernels: Rise of the virtual library operating system. *Queue*, 11(11):30, 2013.

MobiSys'17 June 19-23, 2017, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-4928-4/17/06.
DOI: http://dx.doi.org/10.1145/3081333.3089307