Fog Computing: Current Research and Future Challenges

Julien Gedeon, Jens Heuschkel, Lin Wang, Max Mühlhäuser
Telecooperation Lab, Technische Universität Darmstadt
Email: {gedeon, heuschkel, wang, max}@tk.tu-darmstadt.de

***Abstract*—Acknowledging the shortcomings of cloud computing, recent research efforts have been devoted to fog computing. Motivated by a rapidly increasing number of devices at the extreme edge of the network that imply the need for timely and local processing, fog computing offers a promising solution to move computational capabilities closer to the data generated by those devices. In this vision paper, we summarize these current research efforts, describe applications where fog computing is beneficial and identify future challenges that remain open to bring fog computing to a breakthrough.**

I. Introduction

Cloud computing infrastructures are the predominant way to store data and perform computations today. Cloud computing offers powerful and reliable infrastructures that are scalable and accessible via flexible pay-as-you-go models. However, with the increasing number of resource-constrained mobile devices at the edge of the network (e.g., mobile phones, connected cars, internet of things (IoT) sensors) that need to offload data and computations, cloud computing creates network bottlenecks. Future applications running on these kinds of devices require ultra-fast processing of data, e.g., for augmented reality applications or real-time event detection. Today's cloud computing infrastructures are unable to fulfill these requirements. Therefore, we can observe a trend in research to move computations away from the cloud and closer to the data sources and their consumers.

Fog computing[1] [3], [4], [5] is a promising research direction in this domain. Compared to the cloud, fog computing offers proximate, small-scale resources that can be instantiated dynamically. Fog infrastructures are located between (mobile) end devices and the cloud in an intermediate layer, as depicted in Figure 1. Most often, this intermediate layer represents the access network (e.g., Wifi routers or cellular base stations) and network middleboxes. Because of this, compared to cloud computing, fog computing can provide context awareness and a better support for user mobility. It is important to note that fog computing infrastructures are heterogeneous, i.e., fog computing infrastructures can be hosted on different kinds of physical devices. Table I summarizes the differences between cloud computing and fog computing.

[1]A similar concept is edge computing [1], [2]. In this paper, we use the term fog computing in general to denote infrastructures that are close to the mobile end devices.

Table I
Comparison between Cloud Computing and Fog Computing

	Cloud Computing	**Fog Computing**
Proximity	low	high
Latency	high	low
Geo-distribution	locally clustered	widespread
Infrastructure	centralized datacenters	decentralized cloudlets
Heterogeneity	low	high
Deployment	fixed, static	dynamic, opportunistic
Virtualization	heavyweight (e.g., VMs)	lightweight (e.g., containers)
Connections	long-thin	short-fat
Access	through core network	typically via 1-hop wireless
Mobility support	limited	yes
Context Awareness	no	yes

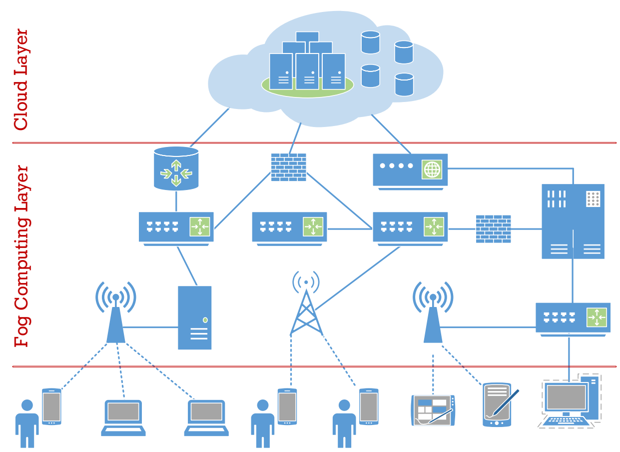

Figure 1. Fog Computing Architecture

In the remainder of this paper, we outline current research topics in fog computing, its possible applications and future challenges.

II. Survey of Current Fog Computing Research

In this section, we review current research efforts in two different domains related to fog computing.

A. *Offloading of Data and Computations*

Mobile device are inherently constrained in terms of computing power and battery lifetime and, thus, there is the need to offload complex computation to more powerful infrastructures. With potentially many fog locations available, the question arises where to place resources and how to allocate them. Making good placement decisions most often requires detailed knowledge wrt. the performance of the offloading system. Meurisch et al. [6] have investigated the probing of unknown services, i.e., how to support the decision on where to offload without detailed prior knowledge of the target system.

While most of the work focuses on the offloading of computations, data offloading is another issue worth considering. Instead of storing data in distant cloud centers, we envision storing data close to where it is used. This is especially relevant if multiple users or applications reuse or share the same data. Of course, this means the storage decision needs to be made in consideration of the current context in which the data is captured. Gedeon et al. present a framework for Android devices that enables context-aware micro-storage of data [7].

B. Fog Computing Infrastructures

Fog computing can be realized on different—mostly already existing—physical infrastructures. One possibility is to collocate computational capacities on the radio access network (RAN), e.g., cellular base stations. Besides implementing fog computing on the RAN, some research also studied the use of privately owned Wifi routers as fog computing devices, either to perform computations [8] or as a mechanism to facilitate service discovery [9]. This is motivated by the fact that these devices are ubiquitously present and often underutilized. Several initiatives already promote free Wifi access (e.g., *freifunk*[2] in Germany). We believe that an open computing ecosystem is the next logical step. Of course, this ecosystem requires new programming models. As an example, Hong et al. [10] have suggested *MobileFog*, a lightweight programming model targeting IoT applications in fog computing.

C. Holistic Resource Management

Achieving efficient operation of fog computing systems is critical, as fog resources are not as abundant as in mega data centers. Ideally, fog nodes should be able to offer compute or storage resources to any user in close proximity through an open and standardized mechanism, which allows a set of fog nodes in the same geographic region to form a shared resource pool. With the help of lightweight virtualization technologies, resources will be allocated holistically at a fine granularity (per user) subject to quality of service and system-wide optimization goals.

Compared to resource management in cloud data centers, resource management in fog computing is more challenging due to the fact that fog nodes are more heterogeneous and uncertainties are imposed by multiple factors such as user mobility. While it is still not available yet, a general centralized framework for holistic fog resource management is envisioned. Based on this assumption, a handful of works have been carried out for fog resource allocation and job scheduling [11], [12], [13], [14], [15], [16], [17]. Jia et al. [12] study the load balancing among multiple fog clouds. Tong et al. [13] discuss workload placement for delay minimization in a hierarchical fog computing architecture. Wang et al. [14] focus on stochastic frameworks for optimizing dynamic workload migration based on Markov Decision Processes (MDPs). Recently, Tan et al. [15] studied online job dispatching and scheduling in fog clouds. Wang et al. investigate online mobility-oblivious resource allocation for fog computing [16] and also develop a service entity placement strategy for social virtual reality applications in the fog environment [17].

[2] https://freifunk.net/

III. Applications for Fog Computing

In this section, we turn our attention to different use cases where we consider fog computing to be beneficial.

A. Internet of Things

The IoT [18] is predicted to grow to billions devices in the upcoming years. According to a recent study by BGC, the predicted market size for the IoT is to reach 267 billion dollars by the year 2020.[3] Comprised of small-scale sensors and actuators, data produced by IoT devices is often consumed only locally. Each of these devices will be delivering massive amounts of data to be used in real-time analytics, event detection or complex event processing. If we consider high-volume data like video, it is obvious that this does not scale. Fog computing however provides the possibility to scale the IoT to a huge number of devices by offering proximate processing of IoT data [4], [19].

B. Smart Cities

An especially useful and palpable usage scenario for fog computing can be found in the vision of *Smart Cities* [20], where urban areas are augmented to provide services to their citizens. This requires to process a multitude of sensor data and distribute it to different actuators. An example is smart traffic management. In this vision, traffic lights would not be programmed statically but adapt their cycle based on different types of data as input. Among others, the data may be provided by inductive loops, video cameras mounted above busy intersections and third-party applications that notify about events, e.g., accidents that have occurred. A comprehensive survey on the implications of fog computing for smart cities can be found in [21]. With the future development of connected cars and autonomous vehicles, quick processing of data to recognize ambient events becomes even more important.

Fog Computing is also interesting for the scenario of emergency response and in crises situations, where other communication infrastructures have become unavailable. In such a scenario, opportunistic infrastructures like smart lamp posts or locally deployed cloudlets can be used to provide disaster relief services [22], [23].

C. Augmented Reality & Virtual Reality

Recently, augmented reality (AR) and virtual reality (VR) applications have gained attention both in research and consumer products. These new classes of applications require ultra fast processing of data, i.e., mostly real-time analysis of video streams. Because even small delays have a considerable

[3] https://www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/

impact on the perceived quality of service, cloud offloading cannot be used. As an example, Ha et al. [24] develop a cognitive assistance application using Google Glasses that allows real time scene interpretation by offloading the computations to VM-based cloudlets. It is worth mentioning that AR and VR applications have other specific challenges related to previously mentioned research, such as the placement of services [17].

IV. Fog-supporting technologies

In this section, we outline how the emerging technologies of lightweight virtualization via cloudlets and SDN/NFV can support fog computing.

A. Cloudlets and Lightweight Virtualization

One prominent concept that has been proposed to contrast cloud computing are *cloudlets* [25], which are micro clouds located at the edge of the network. Cloudlets therefore can run on a variety of devices, including the ones with constrained resources such as network routers. Fog computing requires new lightweight virtualization techniques in order to provide quick provisioning and migration of services on heterogeneous resources. The latter is motivated by the high mobility of users at the edge of the network. In this domain, a lot of research has investigated the use of container-based virtualization, such as *Docker*[4], focusing on migration [26] or adapting Docker for the provisioning of resources at the edge of the network [27]. Recently, library operating systems such as unikernels[5] have received quite some attention as an alternative to the VM-based virtualization technology. Unikernels can be very help in edge computing due to the fact that they are lightweight and are much more secure than containers.

B. SDN and NVF

Software Defined Networking (SDN) splits up the data plane and control plane of networks and is a technology that has recently received a lot of attention. In the context of fog computing, SDN can be leveraged to facilitate the management and organization of networks. Instead of configuring every device individually, a set of rules can be managed and installed by a centralized controller software, leveraging a (potentially) global view on the network. Hence, it is possible to plan and optimize the network traffic even in big and complex systems like the core network of Internet Service Providers (ISPs) [28] by executing these rules on the forwarding devices.

Along with SDN, network function virtualization (NFV) plays a big role in upcoming ISP networks. Virtualization of network functions gives the well-known cloud advantages, like dynamic scaling and better cost efficiency, since special hardware is replaced through cheap commercial off-the-shelf (COTS) servers where virtual network functions can be orchestrated within seconds. However, the usage of COTS servers comes at a price since they arise as bottleneck for 1) network-intensive tasks (simple operations on a huge amount of packets) and 2) compute-intensive tasks (complex functions on a set of data) [29]. To solve this problem, researchers can leverage domain-specific hardware, such as field-programmable gate arrays (FPGAs), to accelerate network functions. Since FPGAs are a very limited resource, they have to be organized in a flexible way. Therefore, Noback et al. proposed a dynamic scheduling scheme for leveraging FGPAs to speed up crucial network functions in an optimized manner [30].

Applications typically choose TCP as the transport protocol, resulting in a big potential for SDN to optimize the network protocol stack by bringing application requirements, transport protocol and link layer in harmony. Heuschkel et al. [31] expanded the SDN paradigm to end devices in order to enable a dynamic control for network protocols. This network protocol virtualization (NPV) approach decouples the applications from specific network protocols, delegating the choice of network protocols to a management instance and enabling the cross-layer optimization of application requirements with the given network environment and available transport layer protocols. To centralize the approach and to give a global view for an end-to-end optimization, Heuschkel et al. [32] proposed an OpenFlow-inspired protocol to communicate management commands, rules and network monitoring information to the end devices. Along with the SDN integration, the NPV approach enables small network functions on end devices, placed as additional layer in the network protocol stack. With these pieces in place, NPV adapts the features of SDN and NFV for end devices, and thus, uses the network stack in a dynamic and optimizable way.

V. Future Challenges

Despite the recent efforts in research outlined before, many challenges still remain open. In particular, we identify the following challenges for future research:

Migration of Data and Applications. Users and devices in most fog computing scenarios are highly mobile. Hence, the services required should also follow these dynamics. This requires the migration of application and data instances across different fog instances. While today this is done mostly reactively, we envision doing this proactively based on the predicted mobility and access patterns of users and data.

Orchestration and Seamless Interplay Between Fog and Cloud. While fog computing undoubtedly offers several benefits and meaningful use cases, we still need cloud infrastructures to persistently store and batch process big data. Data gathered at the edge of the network might be both interesting for the immediate processing as close as possible, but also for cloud applications. Orchestration between cloud and fog services therefore is necessary.

Business Models. The term fog computing was initially coined by Cisco in order to promote their IOx platform. From this history we can see the importance of fog computing as a business opportunity for manufacturers of networking hardware, who can rent out parts of their devices' capabilities

[4]https://www.docker.com/
[5]http://unikernel.org

for general-purpose computing. However, this requires new business and pricing models that capture the cooperative nature of fog computing, as migration of data and services needs to be feasible across the domains of different stakeholders. This becomes even more important when users are competing for resources.

Security and Privacy. If we envision executing container-based applications from different application providers on existing infrastructures, this has obvious security implications, such as third party applications maliciously interfering with the infrastructure. Therefore, strong sandboxing/isolation mechanisms are required if we expect fog computing to be an open ecosystem. Several other security challenges in fog computing have been outlined by Stojmenovic et al. [33]. Because fog computing processes the data close to where it originates, it offers the possibility to employ privacy-preserving mechanisms early in the processing chain. Imagine for instance a video camera stream that is fed as a raw data source to an application that detects the presence of objects on a city street. For this particular application, the faces of pedestrians and license plates of cars are not required and are a threat to one's privacy. Fog computing would allow the blurring of these elements before forwarding the data stream to applications. Application models that can give these kinds of privacy are not yet present in today's fog landscape.

VI. Conclusion

In this paper, we have outlined the current research efforts towards fog computing. We further described applications where fog computing can complement existing cloud infrastructures and identified future challenges that need to be addressed for the widespread adoption of fog computing.

Acknowledgement

This work has been funded by the German Research Foundation (DFG) as part of the Collaborative Research Center (CRC) 1053 - MAKI.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *Proc. INFOCOM Workshop*, 2014, pp. 346–351.

[3] L. M. Vaquero and L. Rodero-Merino, "Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and its Role in the Internet of Things," *Proc. MCC Workshop*, pp. 13–16, 2012.

[5] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and Applications," in *Proc. HotWeb*, 2015, pp. 73–78.

[6] C. Meurisch, J. Gedeon, T. A. B. Nguyen, F. Kaup, and M. Mühlhäuser, "Decision support for computational offloading by probing unknown services," in *Proc. ICCCN*, July 2017, pp. 1–9.

[7] J. Gedeon, N. Himmelmann, P. Felka, F. Herrlich, M. Stein, and M. Mühlhäuser, "vStore: A Context-Aware Framework for Mobile Micro-Storage at the Edge," in *Proc. MobiCASE*, 2018, pp. 1–18.

[8] C. Meurisch, A. Seeliger, B. Schmidt, I. Schweizer, F. Kaup, and M. Mühlhäuser, "Upgrading wireless home routers for enabling large-scale deployment of cloudlets," in *Proc. MobiCASE*, 2015, pp. 12–29.

[9] J. Gedeon, C. Meurisch, D. Bhat, M. Stein, L. Wang, and M. Mühlhäuser, "Router-based brokering for surrogate discovery in edge computing," in *Proc. ICDCS Workshops*, 2017, pp. 145–150.

[10] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proc. MCC Workshop*, 2013, pp. 15–20.

[11] L. Wang, L. Jiao, D. Kliazovich, and P. Bouvry, "Reconciling task assignment and scheduling in mobile edge clouds," in *Proc. ICNP*, 2016, pp. 1–6.

[12] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Proc. INFOCOM*, 2016.

[13] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. INFOCOM*, 2016.

[14] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. S. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Networking*, 2015.

[15] H. Tan, Z. Han, X.-A. Li, and F. C. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proc. INFOCOM*, 2017.

[16] L. Wang, L. Jiao, J. Li, and M. Mühlhäuser, "Online resource allocation for arbitrary user mobility in distributed edge clouds," in *Proc. ICDCS*, June 2017, pp. 1281–1290.

[17] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in *Proc. INFOCOM*, 2018.

[18] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.

[19] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug 2016.

[20] J. M. Schleicher, M. Vögler, S. Dustdar, and C. Inzinger, "Enabling a smart city application ecosystem: Requirements and architectural aspects," *IEEE Internet Computing*, vol. 20, no. 2, pp. 58–65, 2016.

[21] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 32:1–32:43, Jun. 2017.

[22] C. Meurisch, T. A. B. Nguyen, J. Gedeon, F. Konhauser, M. Schmittner, S. Niemczyk, S. Wullkotte, and M. Mühlhäuser, "Upgrading wireless home routers as emergency cloudlet and secure DTN communication bridge," in *Proc. ICCCN*, 2017, pp. 1–2.

[23] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 40–49, Oct 2013.

[24] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. MobiSys*, 2014, pp. 68–81.

[25] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[26] L. Ma, S.Yi, and Q. Li, "Efficient Service Handoff Across Edge Servers via Docker Container Migration," in *Proc. SEC*, 2017.

[27] P. Liu, D. Willis, and S. Banerjee, "Paradrop: Enabling lightweight multi-tenancy at the network's extreme edge," in *Proc. SEC*, 2016, pp. 1–13.

[28] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking : A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14 – 76, 2015.

[29] Z. Bronstein, E. Roch, J. Xia, and A. Molkho, "Uniform handling and abstraction of nfv hardware accelerators," *IEEE Network*, vol. 29, no. 3, pp. 22–29, 2015.

[30] L. Nobach, B. Rudolph, and D. Hausheer, "Benefits of conditional FPGA provisioning for virtualized network functions," in *Proc. Netsys*, 2017, pp. 1–6.

[31] J. Heuschkel, I. Schweizer, and M. Mühlhäuser, "Virtualstack: A framework for protocol stack virtualization at the edge," in *Proc. LCN*, 2015, pp. 386–389.

[32] J. Heuschkel, M. Stein, L. Wang, and M. Mühlhäuser, "Beyond the core: Enabling software-defined control at the network edge," in *Proc. Netsys*, 2017, pp. 1–6.

[33] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurr. Comput.: Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, 2016.