Fog and Cloud Computing Optimization in Mobile IoT Environments

José Carlos Ribeiro Vieira
josecarlosvieira@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa
Advisors: Prof. António Manuel Raminhos Cordeiro Grilo
Prof. João Coelho Garcia

Abstract. We introduce a xxxxx

Keywords: Cloud computing, fog computing, mobility, optimization, multi-objective

Table of Contents

1 Introduction

Cloud computing is a computing technology that became popular at the beginning of the twenty-first century, which provides users online access to services, employing large groups of computers, servers, disks, and routers interlinked together in a distributed and complex manner. Cloud computing has been imperative in expanding the reach and capabilities of computing, storage, data management, and networking infrastructure to the applications. The key idea in this model is that clients outsource the allocation and management of resources (hardware or software) that they rely upon to the cloud. Clouds can provide different service models according to the users' requirements, like infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). Since the demand for cloud resources will change over time, setting a fixed amount of resources results in either over- or under-provisioning, so cloud service providers (CSPs) afford dynamic resources for a scalable workload, applying a pay-as-you-go cost model where clients only pay for the amount of resources they actually use. Cloud computing brings many advantages to the end user applications like high availability, flexibility, scalability, reliability, to mention a few.

Although cloud computing has brought forth many advantages, it has certain limitations. Since cloud servers reside in remote data centers, end-to-end communication may have long delays (characteristic of multi-hops transmissions over the Internet), so the time required to access cloud-based services may not be suitable for some applications with ultra-low latency requirements (real-time). Augmented reality applications that use head-tracked systems, for example, require end-to-end latencies to be less than 16 ms [1]. Cloud-based virtual desktop applications require end-to-end latency below 60 ms if they are to match QoS of local execution [2]. Remotely rendered video conference, on the other hand, demand end-to-end latency below 150 ms [3].

Despite the fact that mobile devices have evolved radically in the last years, battery life, computation and storage capacity remain limited. This means that, heavy application executions must be offloaded to cloud servers, which then return processed results. The solution that has already been proposed is to bring the cloud closer to the end users, where entities such as base-stations would host smaller sized clouds. This idea has been variously termed as Cloudlets [4], Fog Computing [5], Edge Computing [6], and Follow Me Cloud [7], to name a few.

Fog computing is a new computing architecture introduced in 2012 by Bonomi et al. [5], that later, in 2015, big companies such as Cisco Systems, ARM Holdings, Dell, Intel, Microsoft, and Princeton University, founded the OpenFog Consortium, to promote interests and development in this field [8]. It aims to enable computing, storage, networking, and data management not only in the cloud, but also along the cloud-to-thing path as data traverses to the cloud (preferably close to the IoT devices). Fog nodes, also known as fog servers or cloudlets (smaller sized clouds with lower computational capacity), can be placed close to IoT source nodes, due to low hardware footprint and low power consumption as near-user edge devices (e.g., small servers, routers, switches, gateways, set-top boxes, access points) or as multitude of mobile devices. This allows latency to be much smaller, through geographical distribution, compared to traditional cloud computing. Nevertheless, cloud is still more suitable than fog for massive data processing. So even though, fog computing has been proposed to grant support for IoT applications, it does not replace the needs of cloud-based services. In fact, fog and cloud complement each other, and one cannot replace the need of the other. Together, they offer services even further optimized. Allowing enhanced capabilities for data aggregation, processing, and storage, where cloudlets are fundamental

to both improving latencies and reducing network traffic to the cloud. Moreover, Internet connectivity is not essential for the fog-based services to work, what means that services can work independently and send necessary updates to the cloud whenever the connection is available [9].

Fog computing will be crucial in a diversity of scenarios. For instance, heterogeneous sensory nodes (e.g., sensors, controllers, actuators) on a self-driving vehicle, are estimated to generate about 1GB data per second [10]. As the number of features grow, the data deluge grows out of control. Moreover, these types of systems, where people's lives depends on it, are hard real-time what means that it is absolutely imperative that all deadlines are met. Offloading tasks to fog nodes will be the best solution, once a big effort in mobility support has been done through the migration of VMs using cloudlets [11]. Also, in this context, Puliafito et al. address three types of applications where fog is required, namely, citizen's healthcare, drones for smart urban surveillance and tourists as time travellers [12], addressing the needs of low latency and mobility support. On top of that, the number of mobile devices are predicted to reach 11.6 billion by 2021, where the subset of IoT ones are expected to be 929 million [13]. With the increasing number of devices sending data to the cloud, leads to and increasing bandwidth usage, which is a bottleneck in terms of response time. To this end, fog computing is a good solution to prevent it.

Despite the benefits that fog promises to offer such as low latency, heterogeneity, scalability and mobility, the current model suffer from some limitations that still require more efforts to overcome them.

Fog servers are supposed to be closer to end devices so its location has to be ubiquitous. Consequently, less powerful than clouds due to the high deployment cost. If many IoT devices make requests to the same fog node at the same time, it will not have enough computational and storage power to give a prompt response. So it raises the question of should a service currently running in one fog node be migrated to another one, and if yes, where? While conceptually simple, it is challenging to make these decisions in an optimal manner. Offloading tasks to the next server appears to be the solution, however, migrate the related data and processing that initially was one-hop away to the end device to a multi-hop away server, will increase network distance. Consequently, increase the end-to-end latency and bandwidth usage by the intermediate links. Besides, this decision still has to take into account the cost for both the client, with respect to the time of migration and computational delay, and the provider, in terms of energy. Ignoring some of these parameters can lead to make incorrect decisions, what will both violate acceptable latency QoS constraints user's application and damage or defeat the credibility of fog computing.

Another limitation, is that fog assumes that cloulets are always fixed. For instance, it does not foresee that a bus could have computational power. A bus could serve as a fog server and provide support for the end devices inside or outside it. The same can be applied to cars that are nowadays getting better in terms of computational power. Moreover, in a near future, unmanned aerial vehicles (UAVs) could be used for the same purpose, providing support for some social phenomena where a large number of people join in the same place as, for example, in sports games, manifestations, etc [14].

Summing up, we intend to tackle two of the current limitations which are, to the best of our knowledge, untreated problems in the literature. One is to provide support mobility in fog computing, not exclusively to the end devices but also to the fog nodes and the other is to achieve multi-objective fog system design.

This research will, therefore, focus on (...)

The remainder of the document is structured as follows. Section 2 xxx. Section 3 xxx.

Section 4 describes xxxx. Section 5 defines the xxx. Finally, Section 6 presents xxxx and Section 7 xxxx.

2 Objectives

This work has the main objective to present

3 Related Work

In this section we will give some contextual information about concepts and techniques that are relevant to our work.

3.1 2015 An architecture for Load Balancing Techniques for Fog Computing Environment

This paper proposes a new architecture based on load balancing algorithm in Fog Computing environment. Users sent the request to the nearby Fog servers which are consistently maintaining the frequently used data. In case, the users don't get the desired resource in nearby Fog Servers who are also communicating with each other, then that request is forwarded to the cloud servers. To implement this load balancing algorithm they use modified Honey Bee Based algorithm.
Nota: load balancing ?

3.2 2012 Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its Genetic Algorithm

For the problem that the energy efficiency of the cloud computing data center is low, they propose a new energy-efficient multi-job scheduling model. To solve this model, they design an encoding and decoding method for the individuals and construct an overall energy efficiency function of the servers. Meanwhile, in order to accelerate the convergent speed of this algorithm and enhance its searching ability, a local search operator is introduced.
Nota: job scheduling ?

3.3 2017 Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching

This paper considers a specific fog computing network consisting of a set of data service operators (DSOs) each of which controls a set of fog nodes (FNs) to provide the required data service to a set of data service subscribers (DSSs). They address the problem of how to allocate the limited computing resources of FNs to all the DSSs to achieve an optimal and stable performance. Therefore, they propose a joint optimization in a distributed fashion. In the framework, they formulate a Stackelberg game to analyze the pricing problem for the DSOs as well as the resource allocation problem for the DSSs.
Nota: resource allocation ?

3.4 2017 Fog computing job scheduling optimization based on bees swarm

This paper proposes a new bio-inspired optimization approach called BeesLife Algorithm (BLA) aimed at addressing the job scheduling problem in the fog computing environment. This approach is based on the optimized distribution of a set of tasks among all the fog computing nodes. The objective is to find an optimal trade-off between CPU execution time and allocated memory required by fog computing services established by mobile users.
Nota: job scheduling ?

3.5 iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments

[15] In this paper they propose a simulator, called iFogSim, to model IoT and Fog environments and measure the impact of resource management techniques in latency, network congestion, energy consumption, and cost.
Nota: Important

3.6 [123] 2018 When clones flock near the fog

Abdelwahab et al. [16] design FogMQ, a self-deploying brokering clones that discover cloud/edge hosting platforms and autonomously migrate clones between them according to self-measured weighted tail end-to-end latency without the need of a central monitoring and control unit, not having to sacrifice computation offloading gain in cloud platforms. Finally, it is simple and requires no change in existing cloud platforms controllers.
Nota: Ver novamente

3.7 [143] 2018 Virtual fog: A virtualization enabled fog computing framework for internet of things

Jianhua Li et al. [17] propose a layered Fog framework to better support IoT applications through virtualization. The virtualization is divided into object virtualization (VOs), network function virtualization and service virtualization. VOs to address the protocol inconsistency (lack of unified networking protocols that leads to exaggerated overhead); Network function virtualization maps standard networking services to VOs, thus, minimize the communication process between consumers and producers by minimizing latency, improving security and scalability; Service virtualization that composes the community and Cloud Apps from various vendors to serve local Fog users with high quality of experience (QoE) but at low cost. At last, Foglets are involved to seamless aggregate multiple independent virtual instances, Fog network infrastructures, and software platforms.
Nota: Ver novamente

3.8 Vehicular fog computing

3.8.1 [145] 2017 Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management

This paper introduces the challenges of handling big data in the IoV environments. The authors emphasize on the role of fog servers and describe a regional cooperative fog computing (CFC) architecture to support IoV applications. The proposed CFC-IoV architecture consists of two layers. The fog layer is a federation of geographically distributed fog servers, a coordinator server, and the cloud servers, whereas the edge layer includes the vehicular ad hoc network (VANET), IoT applications, and mobile cellular networks.
Nota: Ver novamente

3.8.2 [149] 2018 Toward efficient content delivery for automated driving services: An edge computing solution

They propose a two-level edge computing architecture for automated driving services in order to make full use of the intelligence at the wireless edge (i.e., base stations and autonomous vehicles) for coordinated content delivery. Then they investigate the research challenges of wireless edge caching and vehicular content sharing. Finally, they propose potential solutions to these challenges and evaluate them using real and synthetic traces.
Nota: Ver novamente

3.8.3 [172] 2016 Vehicular fog computing: A viewpoint of vehicles as the infrastructures

Xueshi Hou et al. [18] present the idea of utilizing vehicles as the infrastructures for communication and computation, named vehicular fog computing (VFC), which is an architecture that utilizes a collaborative multitude of end-user clients or near-user edge devices to carry out communication and computation, based on better utilization of individual communication and computational resources of each vehicle. They discussed four types of scenarios of moving and parked vehicles or congested traffic. Also, they point out the advantages against vehicular cloud computing (VCC) and the advantages in scenarios like of emergency operations for natural disaster and terrorist attack.

3.8.4 [175] 2017 Federations of connected things for delay-sensitive iot services in 5g environments

This study looks at the latency issues that may be experienced by delay-sensitive IoT applications due to the geographical distances between the physical IoT devices and the data centers. The authors consider the mobile IoT federation as a service (MIFaaS) paradigm that leverages the pool of devices managed by individual cloud providers as a whole in order to help support delay-sensitive applications. The network model considered is a cellular IoT environment with multiple LTE femtocells as the network edge nodes that supports the MIFaaS paradigm.
Nota: Ver novamente

3.8.5 [177] 2017 Foud: integrating fog and cloud for 5g-enabled v2g networks

This paper argues that cloud and fog computing using the current mobile networks may not be ideally suited to provide the desired levels of QoS for moving electric vehicles in vehicle-to-grid (V2G) services. They propose a hybrid computing model called "Foud" in which the cloud and fog come together and are made available to the V2G networks in 5G. In the proposed model, the cloud allows virtualized computing, storage, and network resources to be available to the V2G system entities, whereas the fog temporarily integrates the stationary and mobile computing resources located at the edge of V2G networks to expand the computing capacity.
Nota: Ver novamente

3.8.6 [183] 2017 Vehicular micro clouds as virtual edge servers for efficient data collection

The concept of vehicular micro clouds based on map information is introduced, and, by a simulation study, investigated. Vehicular micro clouds are virtual edge servers and are essentially clusters of cars that help to aggregate and preprocess data that is transferred to the cloud.
Nota: Ver novamente

3.8.7 [279] 2015 Software defined networking-based vehicular adhoc network with fog computing

Vehicular ad hoc networks (VANETs) face many issues such as unreliable connectivity, delay constraints, and poor scalability. The authors suggest that using principles of fog computing along with SDN could solve many of the current problems with VANETs. In their architecture, the vehicles, which act as end-users, are SDN wireless nodes. These wireless nodes send their data to Road Side Units (RSUs) which are installed alongside road systems. Once the data is sent to an RSU, it is then sent to an RSU controller (RSUC) which is a cluster of RSUs connected by broadband. The RSUC is capable of data storage and processing. Finally, the RSUCs communicate with the SDN controller, which has global knowledge of the VANET system.
Nota: Ver novamente

3.8.8 [354] 2017 A vehicle-based edge computing platform for transit and human mobility analytics

Introduces a Wi-Fi-based in-bus monitoring and tracking system that observes mobile devices and provides analytics about people both within and outside the vehicle. The system can further use the data that is collected by the vehicle-mounted wireless device to track passenger movements, detect pedestrian flows, and evaluate how external factors impact human mobility, which provides useful analytics to transit operators.
Nota: Ver novamente

3.8.9 [365] 2018 Mobile edge computing for big-data-enabled electric vehicle charging

This paper proposes a MEC-based system enabled by big data analytics for the EV charging use case.
Nota: Ver novamente

3.8.10 [379] 2018 Sttr: A system for tracking all vehicles all the time at the edge of the network

They propose a radically different approach to smart surveillance for vehicle tracking. Specifically, this paper explores a smart camera surveillance system aimed at tracking all vehicles in real time.
Nota: Ver novamente

3.8.11 2017 Fog Vehicular Computing Augmentation of Fog Computing Using Vehicular Cloud Computing

In accordance with the resource restrictions of fog computing, only a limited number of clients are able to use fog computing simultaneously. To alleviate this problem, they presented FVC as a new concept in which many unused resources of vehicles can be leveraged to augment fog computing resources. They also depicted the cross-layer architecture for FVC and elucidated its constitutive components along with their role in the construction of FVC. **Nota:** Ver novamente

3.9 [186] 2018 Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning

Unmanned aerial vehicles (UAVs) have been considered as means to provide computing capabilities [14]. In this model, UAVs act as fog nodes and provide computing capabilities with enhanced coverage for IoT nodes. The system aims at minimizing the total mobile energy consumption while satisfying QoS requirements of the offloaded mobile application. This architecture is based on a UAV-mounted cloudlet which provides the offloading opportunities to multiple static mobile devices. They aim to minimize the mobile energy consumption, while satisfying QoS requirements and optimize UAV's trajectory.
Nota: boa introdução, apresenta modelo energético, serirá para referencias introdutórias

3.10 [217] 2017 Mobile edge cloud network design optimization

Alberto Ceselli et al. [19] present a link-path formulations supported by heuristics to compute solutions in reasonable time to firstly determining where to install cloudlet facilities and secondly assigning sets of access points, such as base stations to cloudlets and lastly supporting VM orchestration and considering partial user mobility information, as well as the satisfaction of service-level agreements. Qualify the advantage in considering mobility for both users and VMs. Compare two VM mobility modes, determining that high preference should be given to live migration and bulk migration seem to be a feasible alternative on delay-stringent tiny-disk services, such as augmented reality support, and only with further relaxation on network constraints. Also, they focus on the potential medium-term planning of an edge cloud network in mobile access networks. They study two design cases: 1) network in a static state 2) network state variations in terms of load and service level, caused by user mobility.
Nota: introdução, ver melhor

3.11 [231] 2015 Dynamic service migration and workload scheduling in edge-clouds

This paper, Rahul Urgaonkar et al. [20] present a model to optimize operational costs while providing rigorous performance guarantees as a sequential decision-making Markov Decision Problem (MDP). This model is different from the traditional solution methods (such as dynamic programming) that require extensive statistical knowledge and are computationally prohibitive. First they establish a decoupling property of the MDP that reduces it to two independent MDPs. Then, using the technique of Lyapunov optimization over renewals they design an online control algorithm that is provably cost-optimal. When the

decoupling property holds, it enables the design of simple online control algorithms that do not require any knowledge of the underlying statistics of the MDPs, yet are provably optimal. This technique was applied to dynamic service migration and workload scheduling.
Nota: introdução, não muito importante - o 232 é melhor

3.12   [232] 2016 Segue: Quality of service aware edge cloud service migration

Wuyang Zhang et al. [21] propose SEGUE, a service that achieves optimal migration decisions by providing a long-term optimal QoS to mobile users. This model arises to overcome the limitations of previous studies that propose a static distance-based Markov Decision Process (MDP) for optimizing migration decisions that although it works, it fails to consider dynamic network and server states. SEGUE is a MDP-based model which incorporates the two dominant factors in making migration decisions: 1) network state, and 2) server state. On top of that SEGUE answers the question of when to recalculate the MDP model, because to short intervals would create heavy overhead, and long intervals may translate into lazy migration. SEGUE adopts a QoS aware scheme to activate the MDP model when a QoS violation is predicted to solve for the when to migrate variable.Two components of SEGUE work together to achieve this. One module monitors network states, server workloads and user mobility and the other is responsible for QoS prediction. This allows SEGUE to avoid unnecessary migration costs and bypass any possible QoS violations.
Nota: introdução, importante

3.13   [237] 2013 Migcep: operator migration for mobility driven distributed complex event processing

Most work studying the placement of operators in such an environment completely disregards the migration costs. However, the mobility of users requires frequent migration of operators, together with possibly large state information, to meet latency restrictions and save bandwidth in the infrastructure. In this papers, Beate Ottenwälder et al. [22] present a placement and migration method for providers of infrastructures that incorporate cloud and fog resources. It ensures application-defined end-to-end latency restrictions and reduces the network utilization by planning the migration ahead of time using predicted mobility patterns. Furthermore, present how the application knowledge of the CEP system can be used to improve current live migration techniques for Virtual Machines (VMs) to reduce the required bandwidth during the migration. First, it allows us to amortize the migration costs by selecting migration targets that ensure a low expected network utilization for a suficiently long time. Second, it allows us to serialize the operator for the migration and migrating parts of the operator a priori in away where unnecessary events are not migrated and bandwidth is reduced.
Nota: usa estatisticas para conseguir calcular a migração antecipadamente

3.14   [246] 2016 Primal: Profit maximization avatar placement for mobile edge computing

User Equipment (UE) are moving in the network, and so the E2E (between UE and its Avatar) may become worse, degrading QoS. The live Avatar migration is triggered to adjust the location of the UE's Avatar. However, the migration process consumes extra resources of the Avatar that may degrade the performance of applications running in the Avatar. By

considering the gain (i.e., the end-to-end delay reduction) and the cost (i.e., the migration overheads) of the live Avatar (a software clone; high performance Virtual Machine (VM) located in a cloudlet) migration, Xiang Sun et al. [23] propose a PRofIt Maximization Avatar pLacement (PRIMAL) strategy for the cloudlet network in order to optimize the trade-off between the migration gain and the migration cost by selectively migrating the Avatars to their optimal locations.

3.15 [247] 2017 Towards efficient edge cloud augmentation for virtual reality mmogs

One problem remains unsolved: how to distribute the work among the user device, the edge clouds, and the center cloud to meet all three requirements especially when users are mobile. Wuyang Zhang et al. [24] propose a hybrid gaming architecture that achieves clever work distribution. It places local view change updates on edge clouds for immediate responses, frame rendering on edge clouds for high bandwidth, and global game state updates on the center cloud for user scalability. In addition, they propose an efficient service placement algorithm based on a Markov decision process. This algorithm dynamically places a user's gaming service on edge clouds while the user moves through different access points taking into account the presence of dynamic network states and server workload states, and user mobility. However, unlike many of the service migration solutions which assumes an ignorable service transition time, they acknowledge that it is impossible to migrate an edge service from one edge to another instantly given the size of a VR game world. Therefore, they propose a mechanism to ensure a new edge cloud is activated when a player connects to the new one. It also co-places multiple users to facilitate game world sharing and reduce the overall migration overhead. Also, they derive optimal solutions and devise efficient heuristic approaches and study different algorithm implementations to speed up the runtime.
Nota: introdução, importante no seguimento de MDP

3.16 [261] 2017 Virtual machine placement for backhaul traffic minimization in fog radio access networks

How to appropriately place virtual machines (VMs) into fog nodes in F-RAN systems is very challenging, and will significantly affect the bandwidth consumption of backhaul links. Thus, Ya-Ju Yu et al. [25] analyse the VM placement problem in fog radio access networks (F-RANs) with the objective to minimize the overall back-haul traffic. The back-haul traffic is incurred in two ways: the VM replication and data transmission to the cloud. When a user connects to a fog node and requests an application service, there is no back-haul bandwidth consumption if the fog node has the application VM. Otherwise, the VM has to be replicated on the fog node, or the request is forwarded to the cloud. They formulate the replica-based VM placement problem by considering the computing and storage of fog nodes, the user service constraint, as well as the edge bandwidth constraint.
Nota: introdução

3.17 [270] 2016 An adaptive cloudlet placement method for mobile applications over gps big data

Introduces the concept of movable cloudlets and explores the problem of how to cost-effectively deploy these movable cloudlets to enhance cloud services for dynamic context-aware mobile applications. To this end, Haolong Xiang et al. [26] propose an adaptive

cloudlet placement (via GPS) method for mobile applications. Specifically, the gathering regions of the mobile devices are identified based on position clustering, and the cloudlet destination locations are confirmed accordingly. Besides, the traces between the origin and destination locations of these mobile cloudlets are also achieved.

Nota: introdução, importante?

3.18 [385] 2017 Myifogsim: A simulator for virtual machine migration in fog computing

[11] An extension of iFogSim to support mobility through migration of VMs between cloudlets.

Nota: importante?

3.19 2016 Scalable Fog Computing with Service Offloading in Bus Networks

[27] In this paper, they leverage the characteristics of buses and propose a scalable fog computing paradigm with servicing offloading in bus networks. The bus fog servers not only provide fog computing services for the mobile users on bus, but also are motivated to accomplish the computation tasks offloaded by roadside cloudlets. By this way, the computing capability of roadside cloudlets is significantly extended. They consider an allocation strategy using genetic algorithm (GA). With this strategy, the roadside cloudlets spend the least cost to offload their computation tasks. Meanwhile, the user experience of mobile users are maintained.

3.20 2017 Fog Computing for the Internet of Mobile Things: issues and challenges

[12] This paper analyse what means provide mobility in Fog computing, the main challanges and three good examples where it is necessary. Also point as future work Proactive vs. reactive service migration, Exploit context information to trigger service migration, Fog federation to enable mobile roaming, Virtualization and migration techniques, Compliance with existing interoperability platforms and Integration with mobile networks towards 5G.

3.21 2018 Dynamic Mobile Cloudlet Clustering for Fog Computing

[28] Fog Computing is one of the solutions for offloading the task of a mobile. However the capability of fog server is still limited due to the high deployment cost. In this paper, is proposed a dynamic mobile cloudlet cluster policy (DMCCP) to use cloudlets as a supplement for the fog server for offloading. The main idea is that by monitoring each mobile device resource amount, the DMCCP system clusters the optimal cloudlet to meet the requests of different tasks from the local mobile device.
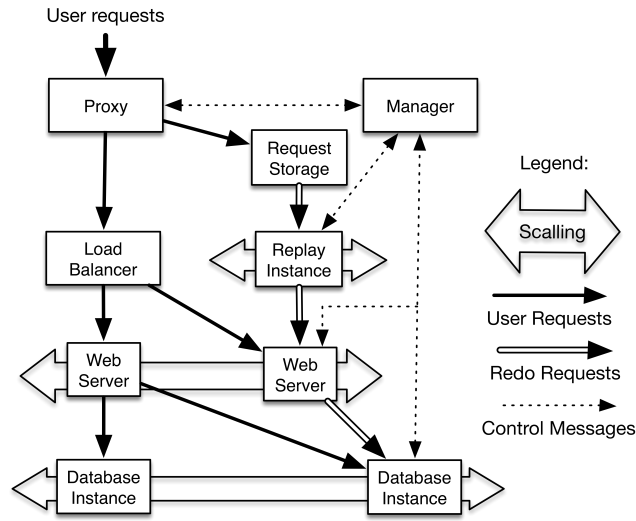
4 Architecture

xxxxx
xxxx

Fig. 1: Overview of the proposed service

5 Evaluation

The evaluation of the proposed architecture will be done xxxx

6 Schedule of Future Work

Future work is scheduled as follows:

– xxxx
– xxxx

7 Conclusion

xxxxx

References

1. S. R. Ellis, K. Mania, B. D. Adelstein, and M. I. Hill, "Generalizeability of latency detection in a variety of virtual environments," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 48, no. 23. SAGE Publications Sage CA: Los Angeles, CA, 2004, pp. 2632–2636.
2. B. Taylor, Y. Abe, A. Dey, M. Satyanarayanan, D. Siewiorek, and A. Smailagic, "Virtual machines for remote computing: Measuring the user experience," *Carnegie Mellon University*, 2015.
3. T. Szigeti and C. Hattingh, *End-to-end qos network design*. Cisco press, 2005.

4. M. Satyanarayanan, "Cloudlets: at the leading edge of cloud-mobile convergence," in *Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures.* ACM, 2013, pp. 1–2.

5. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing.* ACM, 2012, pp. 13–16.

6. S. Davy, J. Famaey, J. Serrat-Fernandez, J. L. Gorricho, A. Miron, M. Dramitinos, P. M. Neves, S. Latré, and E. Goshen, "Challenges to support edge-as-a-service," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 132–139, 2014.

7. T. Taleb and A. Ksentini, "Follow me cloud: interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

8. J. MSV, "Is fog computing the next big thing in internet of things?" Apr 2016. [Online]. Available: https://www.forbes.com/sites/janakirammsv/2016/04/18/is-fog-computing-the-next-big-thing-in-internet-of-things/#6bc7b2c4608d

9. A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *arXiv preprint arXiv:1808.05283*, 2018.

10. A. Angelica, "Google's self-driving car gathers nearly 1gb/sec," 2013. [Online]. Available: http://www.kurzweilai.net/googles-self-driving-car-gathers-nearly-1-gbsec

11. M. M. Lopes, W. A. Higashino, M. A. Capretz, and L. F. Bittencourt, "Myifogsim: A simulator for virtual machine migration in fog computing," in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing.* ACM, 2017, pp. 47–52.

12. C. Puliafito, E. Mingozzi, and G. Anastasi, "Fog computing for the internet of mobile things: issues and challenges," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–6.

13. "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper," Mar 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

14. S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

15. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

16. S. Abdelwahab, S. Zhang, A. Greenacre, K. Ovesen, K. Bergman, and B. Hamdaoui, "When clones flock near the fog," *IEEE Internet of Things Journal*, 2018.

17. J. Li, J. Jin, D. Yuan, and H. Zhang, "Virtual fog: A virtualization enabled fog computing framework for internet of things," *IEEE Internet Things J*, vol. 5, pp. 121–131, 2018.

18. X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

19. A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 3, pp. 1818–1831, 2017.

20. R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Performance Evaluation*, vol. 91, pp. 205–228, 2015.

21. W. Zhang, Y. Hu, Y. Zhang, and D. Raychaudhuri, "Segue: Quality of service aware edge cloud service migration," in *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on.* IEEE, 2016, pp. 344–351.

22. B. Ottenwälder, B. Koldehofe, K. Rothermel, and U. Ramachandran, "Migcep: operator migration for mobility driven distributed complex event processing," in *Proceedings of the 7th ACM international conference on Distributed event-based systems.* ACM, 2013, pp. 183–194.

23. X. Sun and N. Ansari, "Primal: Profit maximization avatar placement for mobile edge computing," in *Communications (ICC), 2016 IEEE International Conference on.* IEEE, 2016, pp. 1–6.

24. W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, "Towards efficient edge cloud augmentation for virtual reality mmogs," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing.* ACM, 2017, p. 8.

25. Y.-J. Yu, T.-C. Chiu, A.-C. Pang, M.-F. Chen, and J. Liu, "Virtual machine placement for backhaul traffic minimization in fog radio access networks," in *Communications (ICC), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–7.

26. H. Xiang, X. Xu, H. Zheng, S. Li, T. Wu, W. Dou, and S. Yu, "An adaptive cloudlet placement method for mobile applications over gps big data," in *Global Communications Conference (GLOBECOM), 2016 IEEE.* IEEE, 2016, pp. 1–6.

27. D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference on.* IEEE, 2016, pp. 247–251.

28. Y. Li, N. T. Anh, A. S. Nooh, K. Ra, and M. Jo, "Dynamic mobile cloudlet clustering for fog computing."