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, wich provide users online accesses 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 end user applications needs, 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 bring many advantages to the end user applications like high availability, flexibility, scalability, reliability, to mention a few.

Although cloud computing has bring 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]. Cloudbased 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, have means that 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 basestations 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. [8]. Later in 2015 big compagnies like Cisco Systems, ARM Holdings, Dell, Intel, Microsoft, and Princeton University, founded the OpenFog Consortium, to promote interests and development in this field [9]. 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 can be placed close to IoT source nodes, due to low hardware footprint and low power consumption (e.g., small servers, routers, switches, gateways, set-top boxes, access points). 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. Fog computing can be seen as a cloud computing extension, namely cloudlets (smaller sized cloud datacenters), located in access points at the edge of the network and hence able to provide lower latencies than the cloud [10]. 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 [2].

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 further efforts to overcome them. We ment to tackle two of the current limitations which are, to the best of our knowledge, untreated problems in the literature. They are the lack of support for mobility of fog nodes and few goals on fog systems design, two closely related issues that are crucial to the proper functioning of fog computing.

The first feature that we want to achieve is provide support for mobility of fog nodes. Most of the existing literature assumes fog nodes are fixed, or focus on the mobility of IoT devices. If fog nodes are mobile, resource availability, offloading, and resources provisioning will be more challenging [11]. As aforementioned, fog computing aims to provide mobility support to users so as they move from one access point to another, data and processing related their device(s) and application(s) move also. This support can be done trougth virtual machines (VMs) migration trougth cloudlets. Comparatible, in cloud computing, migration of VMs is important to support load balancing, power efficiency, fault tolerance, and system maintenance [16], in fog computing is also important to support user mobility. Although fog computing aims to support IoT, without mobility support, fog will have a lot of applications that will no have support. For instansce, heterogeneous sensory nodes (sensors, controllers, actuators, etc.) on a driverless/authonomous vehicule, are estimated to generate about 1 GB data per second [12]. As the number of features grow, the data deluge grows out of control. Moreover, this type of systems, where peoples' life depends on it, are hard real-time what means that it is abuslutly imperative that all deadlines are met. Offloading tasks to fog nodes will be the best solution however it is necessary to handle mobility so VMs can be migrated while the car moves. Also in this context, Puliafito et al. address three types os applications where mobility is also required, namely, Citizen's healthcare, Drones for smart urban surveillance and Tourists as time travelers [13]. On top of all this, the number of mobile devices are predicted to reach 11.6 billion by 2021, where the subset of IoT ones are expected to become 929 million in the same year [14]. Zhang et al. shows another example of application where it is referred the Massively Multiplayer Online Games (MMOGs) and Virtual Reality (VR) technologies, VR-MMOGs and the challanges associated with mobility and others[15]. As the above mentioned, there existis plenty of diferent ones so computing and data capacities should be maintained close to end devices to keep latencies as low as possible and mobility should be provided through VM migration. In this field, there already exists some efforts to this need, however mobility support to the IoT devices will not be suficient to achieve the necessary QoS and QoE that users need. As aforementioned, the number of devices is growing up and support at fog nodes will be essential. Cloudlets are less powerfull than clouds and if a large number of IoT devices make requests to a single one fog node, it will not have enought computational and storage power, so it apears the need of ofloading tasks from the cloudlets where there is litle research and development.

The second feature is multi-objective fog system design. Many schemes (e.g., offloading, load balancing) consider few objectives and ignore other objectives [11]. This feature is closely related to the first one because, it raises the question of should a service currently running in one cloudlet be migrated as the user moves, and if yes, where? This question stems from the basic tradeoff between the cost of service migration vs. the reduction in network overhead and latency for users that can be achieved after migration [17]. On top of that, another question that arises is, should a fog node be moved to provide offloading support to a

overloaded cloudlet and leave the current area with less computing and storage capabilities, and if yes, where should it be placed? While conceptually simple, it is challenging to make this decisions in an optimal manner. This kind of decisions fit perfectly in this system design.

When a mobile client initially secures a one-hop away edge cloud server to ensure the shortest network delay, client mobility may cause the server to be be multi-hops away. The increased network distance, and the potential bottleneck bandwidth that might be introduced by the intermediate links may result in poor connectivity to the cloud service. Even when a mobile user moves around the originally connected edge cloud, service latency may increase because of unexpected crowds of mobile clients seeking to connect to the same edge cloud simultaneously. Thus, increased network or server processing delays may violate acceptable latency QoS constraints.

Cloud service migration may effectively provide expected QoS with respect to user mobility, dynamic networks and varying edge cloud states. To date, previous CloudNet [7] and VM Handoff [8] studies introduced virtual machine (VM) migration in real time under the assumption that the allimportant variables of when and where to migrate were known. These assumptions cannot be made in the real world for two reasons. First of all, conditions that may or may not trigger migration of a cloud service may vary widely. One central consideration that must be accounted for is the tradeoff between the cost of migration and any real QoS improvement. Secondly, we need to quantify long-term performance of cloud servers with respect to any requested service migration to ensure that the best server is chosen, wherein that best choice is realized by the maximization of promised QoS for any mobile user over time. Previous work [9], [10] proposed a static distance-based MDP model that solved the problem of where to migrate by defining each edge cloud migration possibility according to hop counts between it and a mobile user. A cost/reward function in MDP may be used to measure the trade-off between migration costs and performance gains, and it may be used to calibrate the long-term performance improvements by each edge server with respect to any prospective mobile client. Therefore, these studies did work to prove the feasibility of applying MDP with respect to the where to migrate decision. But the static distance-based MDP models did not fully support real-time mobile applications due to its inherent limitation to reflect the two ruling factors in any where to migration decision: 1)network state, and 2) server state. The inherent limitation of previous distance-based models to reflect the two ruling factors in any where to migration decision can be illustrated in the fairly common instance of two edge clouds deemed identical because they both share an equal hop count to a mobile user. Yet, no two edge clouds are ever precisely identical because they are characterized by different network delays and different server processing delays. These differences provide a real difference in the choice between one edge cloud and another of equal hop count. For example, one edge cloud may become heavily loaded and congested because it has the smallest hop count for the client, and, accordingly, be the chosen destination for a specific migration. Another unaddressed problem exists, because a previous MDP edge cloud service migration model recalculates optimal edge cloud migration for mobile users without specifying an optimal interval period for such recalculation. Since running MDP is a computing intensive task, short recalculation intervals introduce the heavy overhead to the server. Conversely, longer recalculation intervals may translate into lazy migration resulting in periods of transgression of QoS guarantees. Finally, of course, operating MDP for edge cloud migration requires realtime feeds of pertinent parameters into the MDP model,

while previous MDP models assume the parameters as static. Such assumptions make these models impractical, if not impossible, to apply to dynamic applications, network states or server states in the real world of real time. [232 - all one needs to know...]

Rather than applying the Fog concept to a specific area, this paper is focused on the realization of Fog.

Simply applying existing radio access-oriented MM (mobility management) schemes leads to poor performance mainly due to the co-provisioning of radio access and computing services of the MEC-enabled (mobile edge computing) BSs (base stations).

More specifically, Fog might be specified in terms of functionality as Fog edge nodes (FENs), Fog server (FS), and Foglet, where FENs and FS are hardware nodes, and Foglet is the middleware in charge of data exchange, as presented in Fig. 1 [6].

1.1 Motivation

The motivation of this paper is to fill the gap by proposing a layered Fog framework to better support IoT applications, encompassing all the layers along the Cloud-to-Things continuum through virtualization. In particular, the virtualization refers to the creation of hardware, operating system, storage device, network resource and event processing by abstraction, orchestration and isolation. In our context, the virtualization is further divided into object virtualization [17], network function virtualization [18], and service virtualization [19].

[17] C. Sarkar et al., "DIAT: A scalable distributed architecture for IoT," IEEE Internet Things J., vol. 2, no. 3, pp. 230–239, Jun. 2015. [18] R. Mijumbi et al., "Network function virtualization: State-of-the-art and research challenges," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 236–262, 1st Quart., 2015. [19] H. Ko, J. Jin, and S. L. Keoh, "Secure service virtualization in IoT by dynamic service dependency verification," IEEE Internet Things J., vol. 3, no. 6, pp. 1006–1014, Dec. 2016.

For example, heterogeneous sensory nodes (sensors, controllers, actuators, etc.) on a driverless car are estimated to generate about 1 GB data per second [3]

A. D. Angelica. Google's Self-Driving Car Gathers Nearly 1 GB/Sec. Accessed: Dec. 6, 2016. [Online]. Available: http://www.kurzweilai.net/ googles-self-driving-car-gathers-nearly-1-gbsec

Starting from available simulators a significant programming effort is required to obtain a simulation tool meeting the actual needs. (deve estar no trabalho a realizar, não na intro..)

1.2 Alternatives

[Alternatives]

1.3 Our approach

To address the aforementioned problems, the present document proposes ...

1.4 Contributions

[Contributions]
The remainder of the document is structured as follows. Section **??** xxx. Section 2 xxx. Section 3 describes xxxx. Section 4 defines the xxx. Finally, Section 5 presents xxxx and Section 6 xxxx.

2 Related Work

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

2.1 xxxx

2.2 xxxx

2.3 xxxx

2.4 xxxx

3 Architecture

xxxxx

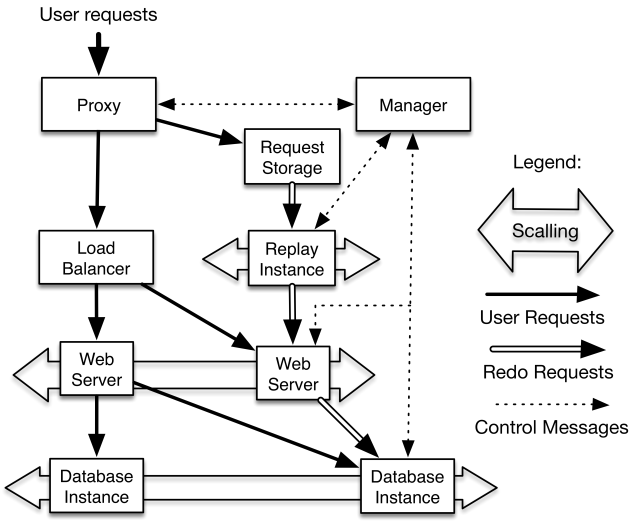

Fig. 1: Overview of the proposed service

xxxx

4 Evaluation

The evaluation of the proposed architecture will be done xxxx

5 Schedule of Future Work

Future work is scheduled as follows:

- xxxx
- xxxx

6 Conclusion

xxxxx

References

1. P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.
2. 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.