2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

Towards Virtual Machine Migration in Fog Computing

Luiz F. Bittencourt*, Márcio Moraes Lopes†, Ioan Petri‡, Omer F. Rana§
*†Institute of Computing, University of Campinas (UNICAMP)
Av. Albert Einstein, 1251, Campinas - SP, 13083-852, Brazil
‡§School of Computer Science and Informatics, Cardiff University, U.K.
Email: *bit@ic.unicamp.br, †marcio@lrc.ic.unicamp.br, ‡petrii@cardiff.ac.uk,§ranaof@cardiff.ac.uk

***Abstract*—Handoff mechanisms allow mobile users to move across multiple wireless access points while maintaining their voice and/or data sessions. A *traditional* handoff process is concerned with smoothly transferring a mobile device session from its current access point (or cell) to a target access point (or cell). These handoff characteristics are sufficient for voice calls and background data transfers, however nowadays many mobile applications are heavily based on data and processing capabilities from the cloud. Such applications, especially those that require greater interactivity, often demand not only a smooth session transfer, but also the maintenance of quality of service requirements that impact a user's experience. In this context, the Fog Computing paradigm arises to overcome delays encountered when applications need low latency to access data or offload processing to the cloud. Fog computing introduces a distributed cloud layer, composed of cloudlets (i.e., "small clouds" with lower computational capacity), between the user and the cloud. Cloudlets allow low latency access to data or processing capabilities, which can be accomplished by offering a VM to the user. An overview of Fog computing is first providing, relating it to general concepts in Cloud-based systems, followed by a general architecture to support virtual machine migration in this emerging paradigm – discussing both the benefits and challenges associated with such migration.**

I. Introduction

There has been increasing reliance on the use of mobile devices to carry out computation and access data. These devices can range in complexity from smartphones, tablet-computers to more complex embedded systems. To fulfil users need for ubiquitous computing capabilities, new paradigms and concepts have become available to offer computational resources anytime and anywhere, among which cloud computing has recently emerged as one of the most prominent computing paradigms to offer on demand computing capacity. Cloud computing has proved successful for a wide range of user needs, offering simple data storage services, on-demand applications, full development platforms, and virtual machines that can be fully managed by the cloud users.

Cloud computing providers offer services through the Internet, where users can have access to services hosted in data centres [1], [2]. Such data centres are often mid to large facilities scattered in a few locations around the globe. This configuration makes users prone to network delays when a service being utilised is not located in a data center geographically close to them. Depending on the applications the user is running, these delays can actually hamper quality of experience when the network cannot fulfil the minimum quality of service requirements of an application, resulting in service degradation/interruption from the user point of view. The limitation posed by the *last mile* connectivity to end users has also been seen as a limitation by other infrastructure providers, such as Content Distribution Network operators (e.g. Akamai and Limelight Networks). These organisations therefore do not fully rely on large scale data centres, as connectivity from the network edge (i.e. user owned devices) may often be the limitation in terms of available bandwidth and latency. Instead, content distribution is often facilitated by specialist servers that are located closer to the user, and which often act as intermediate gateways to channel and manage (e.g. cache) popular content.

To improve service provisioning in the scenario depicted above, it is necessary to reduce network delays in order to consequently reduce quality of service degradation and improve user quality of experience. One way to achieve reduced network delays is to bring service provisioning closer to the user, whilst still maintaining user mobility. Unfortunately, with the current deployment and location of data centres, it is not possible to offer low network delays ubiquitously, thus a new paradigm is necessary to overcome this problem. In this context, the *Fog Computing* paradigm has arisen to overcome such limitations and act as a complement to the current cloud computing paradigm [3].

The Fog computing architecture is based on small distributed data centres, named *cloudlets*, aimed to reduce application delays and/or processing turnaround times, specially for mobile clients [4]. Cloudlets have lower computing capacity and are closer to users, but they can also rely on larger cloud data centres whenever necessary. This approach to multiple data centres (with varying types of capability), which are hierarchically organised, has implications on how users' data and processing are managed to improve service quality as well as reduce costs, bringing new challenges to be tackled. In this paper we provide an overview of Fog computing, discussing the main aspects involved in the interaction among cloudlets and cloud data centres and highlighting benefits of Fog computing. Moreover, we present a Fog computing architecture to provide the management components that are necessary to the smooth operation of *the Fog*, mainly the migration of user's data among cloudlets to maintain quality of service, and the challenges associated with its operation.

The remainder of this paper is organised as follows. Section II presents basic concepts of cloud, mobile cloud, and Fog computing. Section III discusses the role of virtualization

978-1-4673-9473-4/15 $31.00 © 2015 IEEE
DOI 10.1109/3PGCIC.2015.85

technologies in the cloud, and how they are useful in Fog computing, while Section IV presents a layered architecture to support the migration of mobile users' data in the Fog. To make the Fog feasible, business models must be implemented, which are briefly discussed in Section V. Related work is presented in Section VI, and Section VII concludes the paper.

II. Cloud, mobile cloud, and Fog

Fog computing does not aim to replace cloud computing, but to fulfil computing needs that cannot be provisioned by current cloud standards, especially due to delay constraints. In this section we discuss three distributed processing paradigms recently developed, namely cloud computing, mobile clouds, and Fog computing. We present their basic concepts in order to better characterise Fog computing architectural requirements, to clarify their differences, and to describe how they can work together to widen the types of applications that can take advantage of data storage and computation offloading without service degradation.

A. Cloud computing

Cloud computing is a distributed computing paradigm that has now achieved wide scale adoption. It provides storage and processing capabilities on-demand by relying on high-capacity data centres accessible through the Internet, thus this setup allows users to access their data and/or applications anywhere an Internet connection is available [1]. Among the main advantages that has made cloud computing popular are the upfront investment avoidance and abstraction of technical details for the user. These are results of a virtualised environment offered though interfaces at different abstraction levels, commonly named "*Something* as a Service", and that can be canonically summarised into three service models: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service).

The management efforts as compared to usual software deployment outside the cloud is shown in Figure 1. At the topmost level, SaaS offers online software deployments developed by providers and accessible by users through the network, who can utilize software features made available by the developer and/or change pre-defined configurations established by the provider. At the middle level, PaaS offers a development framework for the cloud user to develop and deploy his/her own application. The user has control over the application features and development, but has no control over the physical infrastructure, and the development framework/tools/software is offered by the cloud provider. At the bottommost level of abstraction, IaaS offers virtual machines as a service, which means the user can lease a "computer" that is remotely accessible with administrative privileges. VMs can be chosen from a set of configurations offered by the cloud provider, specifying CPU speed, amount of RAM, amount of disk space, I/O speed, network connection speed, etc. Therefore, in this model users can expand their computational power as necessary, since VMs can be leased and released on demand and with the desired hardware configurations. In practice, this generally implies lower upfront capital investment because one does not need to acquire physical servers to fulfil peaks in demand (which, by definition, are likely to be rare events), which can be fulfilled by renting VMs as the demand grows. When utilising an

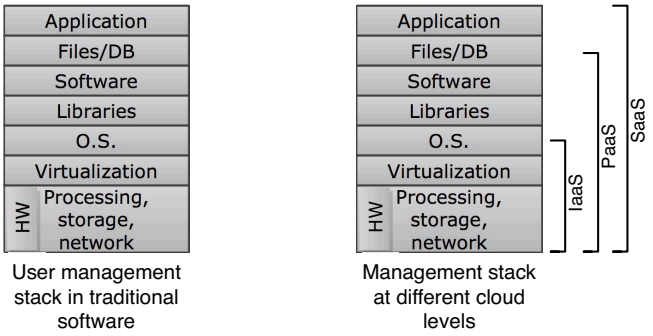

Fig. 1. Different service models in cloud computing and their level of management abstraction (from [5]).

IaaS provider, the user can be charged according to the VM configuration chosen and duration of the rental.

Another IaaS VM rental characteristic that must be considered by the user is the charging model, usually specified in a Service Level Agreement (SLA) that defines how VM rentals are charged and which conditions apply to the specified rental. Popular models include:

i. **On-demand**: the user simply chooses the VM to be rented according to his/her needs and is charged as long as the VM is still running;

ii. **Reserved**: the user pays a fixed fee for the right to rent an on-demand VM but for a lower price;

iii. **Spot**: VM prices are based on the market offer/demand, and the user establishes a maximum price (bid) he/she is willing to pay for that type of VM. The user can have this VM as long as the price does not surpasses his/her bid, and if it does, the VM can be interrupted by the provider without user acknowledgement.

iv. **Auctions**: VMs are auctioned based on demand – with the price reflecting demand for particular types of VMs. This type of market model is particularly prevalent when specialist VMs (e.g. with GPUs or supporting specialist software libraries) are made available to users.

In general, on-demand VMs are more expensive than reserved VMs, and spot VMs are cheaper due to the risk of interruption. The charging model to be chosen is highly dependent on the application requirements and demand predictability. For example, if the user knows a VM will be needed for a whole year and it cannot be interrupted, this VM could be pre-reserved. On the other hand, if the VM is needed for a short term (e.g. a few days), on-demand VMs can result in lower bills due to reduced upfront costs. However, if an application (or, in this case, its hosting VM) can be interrupted, a spot instance would suffice and result in lower bills.

The cloud computing paradigm suits well a variety of applications, specially ones that require ubiquitous data/ processing availability. Users can thus have access to their application and data at different locations across multiple platforms (desktops, laptops, smartphones, smartwatches, etc). With the plethora

of mobile devices and sensors appearing nowadays, cloud computing helps to support processing/ storage needs of low-capacity devices, which also need to save battery. In this context, the mobile cloud paradigm arises, as we briefly introduce in the next section.

There has been recent interest in developing "software defined environments" – an approach enabling dynamic management of various layers of a cloud system. This can range from dynamically changing: (i) number of VM instances & types hosted at a data centre; (ii) network capability (through use of programmable network components and network function virtualisation) and connectivity; (iii) storage instances and their types; (iv) network-based services, e.g. firewalls. This vision of a software defined data centre/enterprise provides significant benefits over current, often provider-based, management of cloud systems.

B. Mobile clouds

Mobile clouds have emerged as an extension of cloud computing capability made available over mobile devices. One central issue in mobile cloud is the decision on what (data and processing) and when to offload from mobile devices to the cloud. Mobile cloud computing leads to various challenges to enable data storage and processing to take place at cloud-hosted computing platforms instead of on mobile devices [6]. Note that a wireless connection is assumed to be available for the mobile cloud to work effectively. Since mobile devices are not always connected or turned on, mobile cloud should take this into account when offloading and caching data in the mobile device to avoid application quality degradation. Various systems have been proposed to achieve this, including support for keeping a replica of a device kernel/memory within a data centre, with periodic synchronisation of state between the device and the data centre. There has also been recent interest in integrating cloud computing approaches with Radio Access Networks (referred to as Cloud RAN) – which aims to provide network function virtualisation at the level of the radio network (e.g. base station broadband processing). Cloud RAN is very much influenced by current developments towards 5G networks (with a focus of significant additional data capacity and energy conservation).

High-demanding applications can be utilised by mobile cloud users, which would not be feasible otherwise due to battery and computing power limitations. Classical examples are image and audio processing applications, which could drain battery and/or take longer to run on a mobile phone than if sent to the cloud. Another important type of application that is enabled by mobile clouds is the data gathering and knowledge extraction resulting from *social clouds* [7] – since socially motivated resource sharing considering the location of multiple mobile clients can allow collaborative applications to support user interests instantly and dynamically.

In devices such as smartphones and smartwatches, mobile cloud is leveraged through applications that hide from the user how data and processing are offloaded. As a consequence, the mobile user may have no knowledge that his/her data or processing is occurring outside the mobile device. In this instance, the user would be unable to tell the difference between a mobile cloud application from a locally running application. From a user's perspective, the mobile cloud may be characterised as SaaS rather than PaaS or IaaS, and the decision on using the cloud for development (PaaS) and/or on-demand capacity expansion (IaaS) is left to the application developer. Therefore, cloud computing provides capacity extension in the form of infrastructure, platform, and software, while the mobile cloud in itself provides capacity extension through software that can use IaaS, PaaS, or even SaaS in the background.

One limitation often observed in the mobile cloud paradigm is the lack of support for low-latency and/or high-bandwidth applications for mobile users, especially interactive applications such as online gaming or augmented reality, but also security-related applications, such as collision avoidance mechanisms for in-vehicle systems, surveillance/tracking with face/object detection mechanisms. This limitation can be overcome by strategic deployment of small data centres, the so-called *cloudlets* [4], [8], closer to the user. To fulfil low-latency requirements and provide a better quality of experience, cloudlets are considered an important aspect of Fog Computing, discussed next.

C. Fog computing

The term *Fog computing* was coined by Bonomi et al. from Cisco [3], [9]. The main feature of the Fog computing paradigm is to extend cloud computing services towards the edge of the network – where the edge consists of mobile devices and cloudlets. In this architecture, mobile devices are connected to cloudlets [4], which are in turn connected to centralised cloud data centres, forming a hierarchical computing platform that has higher computing power at the data centre and lower computing power at the devices. This setup is illustrated in Figure 2. Cloudlets can be located in many places, such as subway stations, cell phone towers, coffee shops, department stores, or even at users home and work, and can be offered to the user according to different business models (see Section V). The connection between user and cloudlets is through a wireless access point and takes place over a single hop communication link. Low latency, data distribution, mobility, and heterogeneity are intrinsic characteristics of the Fog.

Although cloud and Fog essentially provide the same service, i.e. remote execution of application and data storage, the Fog is closer to the user and more densely distributed. Therefore, the Fog supports lower latency applications for mobile users through its geographical distribution, providing faster response for applications and their users. Furthermore, being at the edge of the network turns Fog into an Internet of Things enabler, allowing rapid processing of sensitive data to take decisions and send commands to actuators, and also facilitates the pre-processing and filtering of large data sets, resulting from tens of billions of connected edge devices [9]. Using the three levels of the processing hierarchy (mobile devices, cloudlets, and cloud data centres), the Fog computing paradigm aims to provide a full range of services that can support most application requirements, from batch processing to low-latency, real-time, and high-demanding applications.

Fog computing research is still at its infancy, and limited literature exists that discusses its characteristics [9], [3], [10],

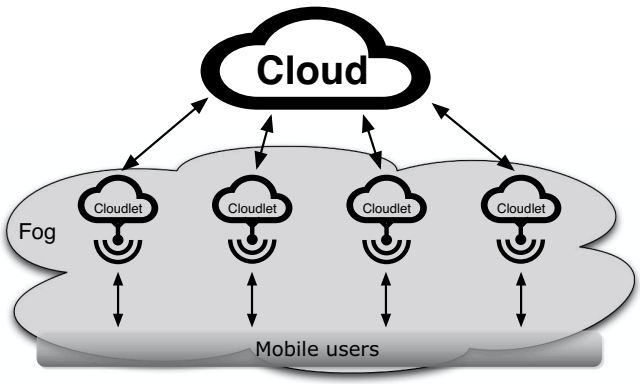

Fig. 2. Fog computing: Cloud, cloudlets, and mobile users work together to provide service to different types of applications.

[11], [12], [13], [14], and no standard architecture or mechanisms to support it has yet been proposed. On the other hand, virtualisation is considered as one of the mechanisms that must be present in the Fog to provide isolation, consolidation, and low-latency support through migration [4]. Support for virtual machines and their migration to offer Fog services are discussed further in this paper. Moreover, although many Fogs may co-exist, independently or in a federation, in this paper we consider a single Fog to discuss peculiarities of the model and related challenges.

III. The role of virtualisation

Although virtualisation is a concept known for decades [15], it remained hidden from the user at lower levels in operating systems mechanisms. In the end of the 1990's, virtualisation gained popularity through hypervisors such as VMWare and Xen, which allow the use of multiple virtual machines to co-exist on physical hardware. With the increase in computing power, system administrators could use virtual machines to consolidate software servers into a few physical servers, maximising utilisation yet maintaining software isolation. As hypervisors evolved, the performance overheads compared to non-virtualised environments have reduced, and virtual machines gained prominence, including those for serving high-demanding applications. In this section we present basic concepts of virtualisation and discuss the main characteristics that make it useful in cloud and Fog computing.

A. Virtualisation in cloud computing

In cloud computing, an important aspect of virtualisation is to provide users with virtual machines that are completely isolated from each other, where user execution environments remain independent. The isolation provided by virtual machines also permits another very important mechanism to be applied: the virtual machine migration. VM migration is the act of transferring the VM and its execution state to another physical server [16], and it can be performed in two different ways:

- **Non-live migration**: the virtual machine is suspended and all its content transferred to another physical machine, the VM is then resumed at the destination host, returning to the same state before it was suspended. In non-live migration, the virtual machine and all its services become unavailable as soon as the suspension starts, and they are only available again after the whole VM is transferred and resumed at the destination. This can take considerable time depending on network conditions and the size of both the virtual machine and the applications data.
- **Live migration**: a snapshot of the virtual machine state is taken and copied to the destination host while the VM is still running. During the copy process, as the VM is running, its state keeps changing. Thus, at the end of the snapshot copying process, *dirty* memory must be copied to overwrite the old memory state present in the last snapshot copied. This is repeated until no (or little) dirty memory exists, or for a pre-defined number of iterations. Then, the VM is suspended in the source host, the remaining dirty memory (if any) is copied to the destination host, and the VM is finally resumed at the destination host, reducing VM downtime. Another strategy for live VM migration is to first suspend the VM in the source host and copy the minimum execution state necessary to resume it at the destination host, and then copy the remaining state after that.

Virtual machine migration in cloud computing is performed mainly to maintain higher system utilisation and to lower energy consumption, as well as for maintenance reasons. When a physical server is underloaded, its virtual machines can be migrated to a server with higher load (but that does not get overloaded after the migration), and the source server can then be put into standby mode, increasing average system utilisation and reducing power consumption.

B. Virtualisation in Fog computing

Virtual machines in clouds provide the user with computing capabilities to extend his/her local computing capacity. In the context of a company, for instance, it can be extending the company's computing cluster capacity because it is overloaded at a given time. For an individual user, it may be extending his/her desktop capacity and/or being used to provide ubiquitous access to data.

In Fog computing, the user also has a cloud-hosted VM. However, another VM (or set of VMs) for the same user can be available at cloudlets, which is generally one (network) hop away from the location of the user, and therefore does not incur a significant network delay. The VM located in the cloudlet does not need to be a full copy of the VM in the cloud, but it has to have the necessary data for the user in a given context/ time. Determining what constitutes "necessary data" is not straightforward, and it involves prediction and monitoring mechanisms to determine what should be closer to the user. A less sophisticated Fog approach can leave this to the application programmer, who uses a Fog programming API to flag important/priority content to be made available in the cloudlets. Hence, if the Fog is able to always have a VM at one hop distance (i.e., in a cloudlet) with all the data its user needs, then the user will minimise delays in offloaded

application execution or data access. Certain challenges need to be addressed to keep this VM always close to the user in a mobile context, where this mobility can be supported through migration of virtual machines among cloudlets, since the user's data and applications should be available at the current user's location. However, the migration process is not trivial, and must take account of data communication delays and minimize the time users spend without (or with delayed) access to their data and/or application.

Bonomi et al. [3] describe Fog Computing as a cloud brought to the edge of the network, providing a number of services for Internet of Things (IoT) and its applications, such as connected vehicles, smart grids, smart cities, and smart wireless devices in general, including sensors and actuators. Besides supporting these applications, we also consider Fog as a provider of services to user applications that require low latency, i.e., interactive applications such as online gaming, real-time disabled aiding applications, or video-interactive streaming. Moreover, bringing user's personal data closer to his/her geographical location can reduce network traffic both in core network and centralised data centres. For instance, once a user stores data in the Fog, the computing infrastructure that makes up the Fog should be responsible for tracking a user and moving data among cloudlets [4], [17]. The centralised cloud data center role will then focus on data storage and high-level management of the Fog, as well as on processing computationally expensive tasks that do not have low delay requirements, such as complex data analytics and big data processing to extract knowledge from data aggregated by the Fog in the IoT context [9], [12].

IV. A Fog architecture to support VM migration

In this work we will consider a Fog computing scenario as illustrated in Figure 3. We are interested in investigating how user's data can be migrated according to his/her mobility [18], aiming to maintain quality levels for applications demanding lower latency than that achieved by using the cloud. The key challenge is understanding how migration could be performed, so that users do not notice performance degradation in their applications, regardless of the type of applications they are running. In Figure 3 we can see a set of mobile device users crossing cloudlet boundaries, requiring the Fog manager to take action to migrate users' data in the direction of their movement – similar to a handoff procedure in cellular networks. Users who are within a cloudlet range, as for example in a park or in traffic, should have their data maintained in the current cloudlet until they cross the migration boundary.

To achieve the migration of user's context, we consider that each user has a virtual machine (VM)[1] running in a cloudlet, as shown feasible by Satyanarayanan et al. in [4]. How and by whom this VM is setup is out of the scope of this paper. In essence, this is a problem yet to be addressed, since the VM (or container) can be offered as a service to the user through his/her mobile operating system, through an application, or directly setup by the user.

In this work, this personal VM is responsible for providing computing capacity (processing/storage), therefore is of

[1]We use the term virtual machine, but the model can be generalised to container-based implementations.

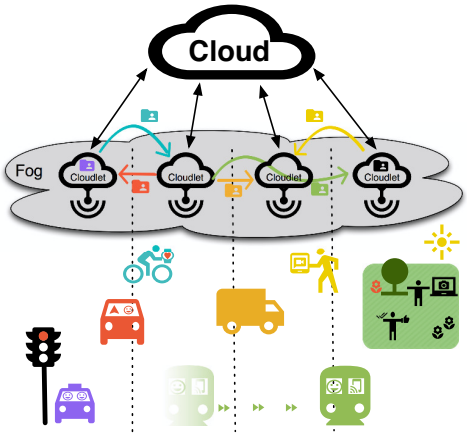

Fig. 3. Fog computing: cloudlets & personal data migration with user mobility.

paramount importance for it to be readily available to its owner. The mechanism that triggers a virtual machine migration is essentially a decision-making process based on an objective function, which is optimised to leverage the best results for both users and service providers. As Fog computing research is still in its infancy, no modelling of this optimisation problem has been proposed in the literature. We argue that such a mechanism is of core importance to efficiently build the Fog and fulfil users needs.

In this section we aim to define general architectural components needed to make location-based virtual machine migration feasible in the Fog computing paradigm. We focus on defining the architecture and discussing its components, interfaces and interactions, along with information needed to support the migration decision-making process (users location, direction/ speed of movement, applications running, amount of stored data, data traffic needs, cloudlets capacity, network capacity, users' usual behaviour, etc). Figure 4 illustrates a layered Fog architecture comprising mobile devices, cloudlets and cloud computing systems.

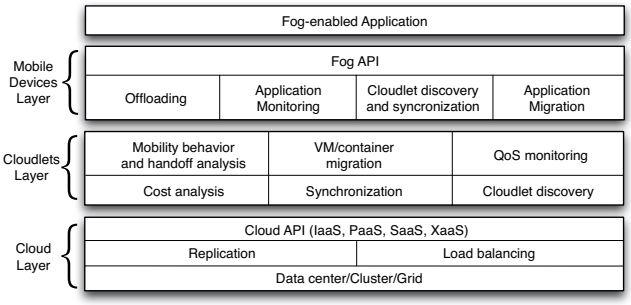

Fig. 4. Fog computing layered architecture.

At the top of the architecture, the Fog-enabled application runs in a mobile device that has the ability to communicate with cloudlets in the lower layer. The application programming interface for Fog supports a set of pre-defined functions, e.g. support for VM migration decision making. Some basic

functionalities that should be supported by this layer through the Fog API are as follows.

[3pt] *a) Data location/offloading*: this functionality should enable the application programmer to decide where to store application data, i.e., leave it on the device or offload to cloudlets or to the cloud. This decision can be based strictly on a priority list according to the availability of each level in the hierarchy. For example, there might be a cloud but not a cloudlet available at a given moment; or there might be no connection available, thus data would be placed on the mobile device until a connection is available. The decision can be also based on measured quality of service parameters, as for example cloudlet processing and storage capacity, or network delays/capacity to cloudlets and to the cloud. Moreover, this functionality can also have parameters that determine if data replication should be available and where it should occur (i.e., in one or several cloudlets in the region, or in the cloud);

b) Processing location/offloading: this functionality is to enable computation offload in a similar way to data replication and with similar parameters. One useful parameter in this functionality would be to attach/detach data location to a function or application. Using this approach, the required data would always be offloaded along with the code to be run. This can bring easier implementation for the application, but background management and delays can reduce application quality;

c) Synchronization: if data is replicated, it must be synchronised. The synchronization functionality should bring option to the programmer to set time intervals for synchronization among replicas, create replicas, and delete replicas. Note that here, at the application programming level, replication and synchronization concerns a single application, and not the entire set of user hosted data. Replication and synchronization at a higher level (i.e., for all the user data) may be decided by cloudlets at a higher level, as we discuss further in this paper, or can be a configuration chosen by the user in his/her fog provider.

d) Cloudlet discovery: find closest cloudlet that is able to comply with given quality of service parameters, which may include network delay to mobile device and to other cloudlets or clouds, processing capacity, storage capacity and speed, and so on.

e) Migration: move application data to user VM in the specified cloudlet, that could be discovered by the cloudlet discovery function or can be a well-known cloudlet for this user.

f) Monitoring: application monitoring to detect degradation and/or behaviours that may trigger one of the actions performed by the functionalities above.

The Fog API enables data and computation offloading and migration control for the application developer. The cloudlet layer offers control over the user's data, i.e., the virtual machine (or container) that hosts the data. The functionalities offered by the cloudlet layer are as follows:

a) Mobility behaviour and handoff analysis: this part of the cloudlet middleware is responsible for detecting user movement and behaviour to take decisions on when and where to perform VM migration. Its implementation can be a simple detection of user movement through disconnection from the cloudlet access point, and migrating the VM along the path the user is following identified by a GPS system or by the user connecting to another cloudlet access points. This reactive mechanism can result in delays in moving the VM if the user has significant amount of data. Predictive techniques may be used to determine where and when the user will move, and then start migrating the VM beforehand, or even synchronising with existing VMs in the expected user path.

b) VM/container migration: this module is responsible for actually performing the migration according to the decisions taken by the mobility behaviour component. This component relies on the assumption that cloudlets have a common migration interface.

c) QoS monitoring: cloudlets can be aware of QoS requirements to check if they are able to comply with user needs. This quality of service can be an aggregation of all application QoS metrics defined in the application layer, and how this aggregation occurs will depend on the type of metric being considered. For example, for considering overall bandwidth, the QoS would be sum of bandwidth requirements of all applications. If a cloudlet detects that it can no longer comply with a users applications QoS, it proactively looks for nearby cloudlets that potentially can.

d) Cost analysis: depending on the business model (discussed in Section V), as well as on the type of cloud being utilised (SaaS, PaaS, IaaS, XaaS, etc), cost analysis must be performed when decisions to offload are taken. For example, if the interaction between cloudlet and cloud occurs at the PaaS level, offloading data or processing requests, or even synchronising user's VM with the cloud, can result in charges to the Fog provider (or to the user, again depending on the business model).

f) Cloudlet discovery: cloudlet discovery at this level aims to build knowledge of the underlying Fog *topology*, to facilitate efficient migration.

Each component of the cloudlet layer raises challenges to be addressed, comprising new algorithms and techniques for both prediction and efficient VM migration. The specific challenges will depend on the objective function(s) defined by the cloudlet provider, and also on the quality of service requirements from its users.

Conversely, the cloud layer implements more mature concepts, such as replication among data centres and load balancing within a data center. It offers service interfaces at different levels (SaaS, PaaS, IaaS, XaaS, etc) for the Fog provider to build the cloudlets layer. The service level of this interaction will depend on what the Fog provider wants to offer to its clients. Based on the set of features to be offered, the Fog provider must build business and charging models to provide SLAs to the final user, as discussed in the next section. Moreover, the interaction between cloudlets and the cloud can make use of existing cloud APIs. To offer Fog services, it is necessary to identify suitable business models for both infrastructure providers and end users – this is the focus of the next section.

V. Business models

Several business models may become relevant when considering virtual machines in the context of Fog computing.

The intermediary layer between the user and the Fog must be put in place and managed by an organisation, and the costs of the cloudlet infrastructure must be taken into account in the business models. Similar to current broad availability of WiFi access points, we envisage three general ways of funding cloudlets: (i) by cloud providers; (ii) by local businesses; (iii) by public funding. We can look into different ways of offering virtual machine support for Fog users:

Service Provider / Service Orientation: in this model, the user would be able to choose a Fog provider on-the-go, according to his/her current activity or provider's availability. The use of a service-based approach enables loose coupling, enabling an eco-system of providers to co-exist. However, there is no guarantee that integrating externally provisioned services will lead to the fulfilment of the user objectives, since this would depend on providers' agreements to support a virtual machine migration across their boundaries.

Support and service contracts: in this model, the user would rely on informative/detailed contracts that adequately capture the circumstances and criteria that influence the performance of the externally provisioned services that are subject of the contract. From the provider side, short-term contracts have proved to be more profitable options for service providers. In performance-based contracts the service provider guarantees a certain level of performance (e.g. availability) of the service to the customer which is reflected in the associated price.

All-in-one enterprise cloud: this business model represents a new approach for providers to organise service offerings and generate revenue. It is a more comprehensive business solution consolidating other business activities and strategies. With this model, cloud providers that offer Fog services can join with local businesses to build a larger business ecosystem with greater financial stability, allowing users' VMs to freely travel across their boundaries.

Understanding the computing business models can help users make informed decisions about which provider to use. Business models come in association with various cost models based on an expected demand/workload profile or determined business objectives. Such cloud cost models are listed below:

(i) *Consumption-based cost model*: where clients only pay for the resources they use. For example, in a Fog where cloudlets are offered by a cloud provider, the user could be charged according to VM size and the number of migrations performed.

(ii) *Subscription-cost pricing model*: where clients pay a subscription charge for using a service for a period of time – typically on a monthly basis. This subscription cost typically provides unlimited usage (subject to some fair use constraints) during the subscription period. This model could be also adopted to provide VM support in the Fog. For example, local businesses can offer a subscription to their infrastructure that enables a user to migrated their cloudlet to this location.

(iii) *Advertising-based cost model*: where a client gets a no-charge or heavily-discounted service whereas the providers receive most of their revenue from advertisers. This model is quite common in cloud-based media services such as free TV provider net2TV, and can also be adopted in the cloudlet context.

(iv) *Market-based cost model* : where a client is charged on a per-unit-time basis. In Fog computing, the user can have a configuration dashboard for cloudlet VMs, establishing its capacity and other relevant parameters, similarly to IaaS offerings such as Amazon EC2. Charging models can, for example, follow the same cloud models described in Section II-A. With millisecond VM offerings from Amazon (e.g. Amazon Lambda), this model becomes viable for mobile users.

(v) *Group buying cost model*: where clients can acquire reduced cost services only if there are enough clients interested in a deal. This can be adapted to Fog if a set of clients agree to share one virtual machine for their data, reducing the Fog provider costs and allowing reduced charges.

VI. Related Work

Fog computing emerged from the need for ubiquitous computing comprised of internet of things (IoT) devices and mobile devices [11]. In this context, Satyanarayanan et al. [4] presented the concept of *cloudlet* as part of a three-layer hierarchical architecture with mobile devices, cloudlets, and cloud computing. Verbelen *et al*.[17] consider that all devices in a LAN can cooperate with one cloudlet, while Stojmenovic [19] states that cloudlets act as an intermediary layer between the cloud and mobile devices aiming to bring services closer to the users. Moreover, some security issues have been risen by Stojmenovic and Wen in [10], where authors emphasise the need for the Fog to provide low latency, location-knowledge, and quality of service for real-time applications.

In [9] the authors state that IoT devices require mobility, geo-distribution, low latency, and a communication network. To bring computational capacity to those IoT devices, which often have constrained capacity, virtual machines can be deployed in order to serve as a capacity extension for those devices. However, depending on the application, VMs in the cloud do not offer the necessary quality of service, specially in what regards to network latencies. Therefore, cloudlets should also support virtualisation to bring this capacity closer to the IoT and mobile devices.

The Cloud of Things [11] is a similar concept to Fog, namely an interaction between IoT and Cloud Computing. The authors focus on how data trimming decisions can be performed by a smart gateway to avoid unnecessary data to be sent to the cloud, matching the concept of cloudlets previously introduced in [3]. A high-level layered architecture for smart gateways is presented.

In [13], Vaquero and Rodero-Merino define Fog as a cloud brought to the edge of the network, being related to the virtualisation of the network infrastructure. Aiming to escape from pitfalls resulted from non-solid definitions or multiply-defined concepts, the authors focus on a more comprehensive definition of Fog computing, which comprises all kinds of devices that could take part of the Fog, and focuses on administratively independent devices composing the fog. In the present work we focus on cloudlets, which are part of the

definition above, appearing as components of a decentralised data center scattered through an area to provide low-latency access to storage and processing capabilities.

In [14], the authors present a programming model for a *Mobile Fog* with the aim to develop IoT applications that take into account their geographical distribution, with large-scale applications which are latency-sensitive. In [20] the authors analyses the performance of live VM migration considering a variety of parameters. Although they do not evaluate a Fog scenario, this evaluation can be useful to determine when (and in which conditions) VM migration must be performed in the context of Fog computing in order to avoid QoS degradation. In [21], Elgazzar et al. propose a support mechanism for computation offloading in mobile services. They propose a system to enable offloaded computation to follow the mobile user and be delivered through the cloudlet that is currently closer to the user. In [22], the authors propose an offloading mechanism from mobile devices to cloudlets, but they do not consider migration among cloudlets.

VII. Conclusion

The ability to more seamlessly integrate data centre/ cloud functionality with a diverse range (of network connectivity, storage and computational capability) of mobile devices remains an important challenge in cloud computing – we provide a comparison of cloud computing, mobile clouds and Fog computing. Moving computation and data storage functionality to the edges of a network makes more effective use of emerging capability in mobile devices. Fog computing proposes the deployment of multiple "cloudlets" which are closer to the user (in terms of network latency), compared to large scale data centres. In this way, as a user moves across multiple locations, a user session (for streaming and gaming applications, for instance) can be handed over to multiple cloudlets, thereby maintaining a constant/ consistent session for the user. We attempt to better characterise properties of Fog computing and its relationship to cloud-based systems – identifying, in particular, how both can be used more effectively together. Virtual Machine (VM) migration remains an important enabling capability in Fog computing, and we focus on how this approach can be used in Fog computing. A set of business models are also proposed which could be used to support revenue generation for various stakeholders in this context. Security concerns (associated with both user data and applications) will remain an important challenge in Fog computing – as cloudlets hosted by third party infrastructure providers will need to be trusted. This is a similar concern that remains in cloud computing – although with the significant increase in the number of potential hosting locations, security will become an even more important concern and research challenge in Fog computing.

Acknowledgment

M.M.L. and L.F.B. would like to thank CAPES and CNPq for the financial support.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Comm. of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] L. F. Bittencourt, E. R. M. Madeira, and N. L. S. Da Fonseca, "Scheduling in hybrid clouds," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 42–47, 2012.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *MCC workshop on mobile cloud computing*. ACM, 2012, pp. 13–16.

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

[5] L. F. Bittencourt, E. R. M. Madeira, and N. L. S. da Fonseca, "Resource management and scheduling," in *Cloud Services, Networking, and Management*. John Wiley & Sons, Inc, 2015, pp. 243–267.

[6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[7] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social cloud: Cloud computing in social networks," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, July 2010, pp. 99–106.

[8] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Fut. Gen. Comp. Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.

[9] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[10] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2014, pp. 1–8.

[11] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Intl. Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2014, pp. 464–470.

[12] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing," in *IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2014, pp. 325–329.

[13] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[14] 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 *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 15–20.

[15] J. E. Smith and R. Nair, "The architecture of virtual machines," *IEEE Computer*, vol. 38, no. 5, pp. 32–38, 2005.

[16] F. Messina, G. Pappalardo, D. Rosaci, and G. M. Sarné, "A trust-based, multi-agent architecture supporting inter-cloud vm migration in iaas federations," in *Internet and Distributed Computing Systems*. Springer, 2014, pp. 74–83.

[17] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: bringing the cloud to the mobile user," in *Workshop on Mobile cloud computing and services*. ACM, 2012, pp. 29–36.

[18] D. Johansson and K. Andersson, "Web-based adaptive application mobility," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*. IEEE, 2012, pp. 87–94.

[19] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Australasian Telecommunication Networks and Applications Conference (ATNAC)*. IEEE, 2014, pp. 117–122.

[20] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster computing*, vol. 16, no. 2, pp. 249–264, 2013.

[21] K. Elgazzar, P. Martin, and H. Hassanein, "Cloud-assisted computation offloading to support mobile services," *Cloud Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[22] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)*, Dec 2014, pp. 109–116.