Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 69 (2014) 704 – 711
Procedia Engineering
www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Cloudlet Challenges

Aleksandar Bahtovski, Marjan Gusev*

Ss. Cyril and Methodius University Faculty of Information Sciences and Computer Engineering
1000 Skopje, Macedonia

Abstract

The vision about having information at our fingertips anytime and anyplace was only a dream in the mid 1990's. Today, Web access, mobile applications and ubiquitous mail is a reality experienced by millions of users worldwide. The main obstacle for today's mobile devices is the lack of resources, since today's applications are becoming more and resource extensive. This requires increasing the processing power of the device, increasing the memory, a need for greater durability of the battery and so on. All of these tendencies would increase the processing speed of applications and would improve the way they function. Therefore, the need for a more powerful device with a better processing speed becomes inevitable. This is a valid reason for introducing cloudlets, which represent intermediate layers located between the cloud and each mobile device. In this way, instead of accessing a distant cloud, the users connect to the nearest cloudlet through a wireless network. The purpose of this article is to explain the importance of using Cloudlets in our everyday life, by presenting the main challenges and possibilities of further research. We present a short introduction about Cloudlets; the cloudlets architecture and the way they operate. Moreover, we discuss about what their features are, how they work with thin and tick clients, and how they operate as offload elements. In the end, we state the challenges and open research question related to the topic.

Selection and peer-review under responsibility of DAAAM International Vienna

Keywords: Cloudlets; Cloud Computing; (VM) Virtual Machines; Offloading;

1. Introduction

The wide availability of Internet access through mobile devices, such as phones, tablets, media players, allows users to access the cloud services while they are on the go. Currently there are more than 90 million mobile Internet

* Corresponding author. Tel.:+389-77-772222
E-mail address: marjangusev@gmail.com

1877-7058
Selection and peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2014.03.045

users. Compared to 2010, the number of mobile Internet users has increased by 50 million. This information indicates that mobile devices are quickly becoming a dominant computing platform. As the capabilities of mobile devices advance (CPU power, memory, connectivity) people increasingly use them for other tasks such as emailing, Internet banking, GPS, gaming, routing etc.

Despite recent advancements in technology, mobile devices are still resource poor. Battery life, weight, insufficient memory and heat dissipation limit the computational resources, and make mobile devices much more constrained than their non-mobile counterparts.

One of the solutions to overcome these resource limitations is mobile cloud computing. By leveraging infrastructure, such as Amazon's EC2 cloud, Microsoft Windows Azure or Rackspace, computationally expensive tasks can be offloaded to the cloud. But, these clouds are far away from mobile users and the high WAN latency makes this approach inefficient for real time applications. To solve these issues, Satyanarayanan introduced the concept of cloudlets [1], as trusted, resource rich computers, located near the mobile users.

In this paper we present the main challenges about cloudlets and possibilities for further research. The paper is organized as follows. Section II gives the background of cloudlets. We explain the motivation for cloudlets in Section II-1, present an overview of the full architecture and the way cloudlets operate in Section II-2, and define the ways in which a cloudlet can be used in Section II-3. Specific features are presented in Section III, including a description of cloudlets as offload elements in Section III-1, then a description how cloudlets work with thin or fat mobile clients in Section III-2 and a description of the usage of VMs in a cloudlet in Section III-3. The main discussion about challenges and open research questions are presented in Section IV. Finally, the last Section gives a conclusion and the next steps in our work.

2. Background

In this section we explain the main motivation for using cloudlets, and present a description of their architecture and the way they work.

2.1. Why Cloudlet?

The number of customers that use a mobile phone to connect to the Internet is constantly rising. It has long been recognized that mobile hardware is necessarily resource-poor relative to static client and server hardware. Mobile devices have more drawbacks compared to static clients and servers. Battery life, memory, weight and heat dissipation limit the computational resources of the device [2].

One of the solutions for increasing the problems about mobile device limitations can be cloud computing. However, WAN delays in the critical path of user interaction can hurt usability by degrading the crispness of system response. Even trivial user-application interaction incur these delays in cloud computing.

That is the reason why cloudlet needs to be implemented. It is a small cloud located nearby mobile users which will be connected through LAN network with clouds located far away [3].

One of the cloudlet benefits is definitely a possibility that mobile users can rapidly instantiate custom virtual machines (VMs) on the cloudlet running the required software in a thin client fashion.

2.2. What is Cloudlet?

Cloudlet is a small cloud located close to the mobile users [4]. It is installed on discoverable, localized, stateless servers running one or more virtual machines (VMs) on which mobile devices can offload expensive computations.

A cloudlet is a set of trusted, resource-rich computers that is well-connected to the Internet and is available for use by nearby mobile devices. Cloudlets do not have to be a fixed infrastructure close to the wireless access point, but instead can be formed in a dynamic way with any device in the LAN network with available resources.

The basic cloud infrastructure is located far from the mobile user. For example, Amazon's EC2 infrastructure is located only in 6 cities worldwide. So when the customer is trying to reach one of these sites from his mobile device, its "end to end" path would include many network segments. High WAN latencies make this approach of direct using the clouds insufficient for real-time applications of mobile users.

Mobile devices are vulnerable on DoS attacks especially when the cloud offload site is located far away. This initiates a motivation to use cloudlets. Mobile users can rapidly instantiate custom virtual machines (VMs) on the cloudlet running the required software in a thin client fashion [5].

There are two important drawbacks of VM based approach to build a cloudlet. In the first drawback all devices in LAN network can cooperate in the cloudlet as shown in Figure 1.

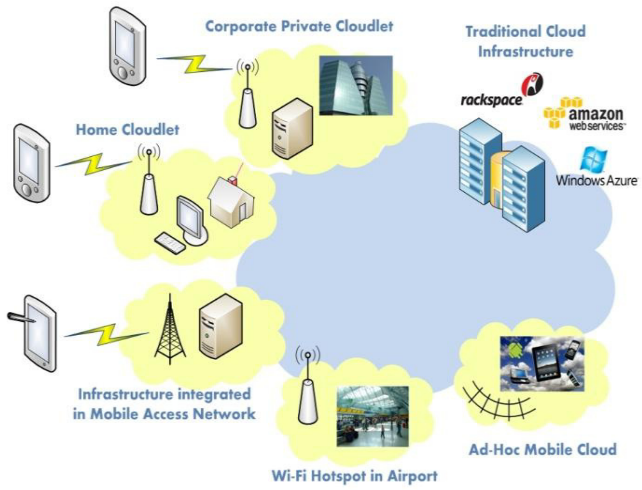

Fig. 1. Types of Cloudlets.

This kind of cloudlets can be implemented in a corporation as a corporate cloudlet, where all devices can share their resources. It can be also implemented in a home network, or mobile network, on the train, etc.

In the second drawback, better performance can be achieved by dynamically partitioning the application in components instead of executing the whole application remotely in the Virtual Machine (VM). A mobile device functions as a thin client and all significant computation occurs in the nearby cloudlet. If no cloudlet is available nearby, the device will try to connect to a distant cloud or in the worst case use its own resources. Full functionality and performance of the device can return later when the device discovers a cloudlet nearby.

Fig. 2. Cloudlet Concept.

Cloudlets are decentralized and widely dispersed Internet infrastructure components whose resources can be leveraged by nearby mobile computers, as illustrated on Figure 2. In this example, a coffee shop integrates nearby mobile devices and communicates with a distant cloud on Internet.

Essentially a cloudlet is a ”data center in a box”. It is self managing, requiring little power, Internet connectivity and access control. Internally cloudlet resembles a cluster of multicore computers with gigabit internal connectivity and a high bandwidth wireless LAN.

Further on, we explain basic approaches used in clouds and cloudlets, and try to explain the related concepts. Table I summarizes some of the differences between cloud and cloudlet. We can conclude that cloudlets are self-managed small data centers, located at business premisses and serve few users at a time. The ownership is decentralized to business users. On the contrary, clouds are professionally administered servers, located on a distance location with special power and cooling conditions and can serve thousands of users.

Table 1. Differences: Cloudlet vs. Cloud

	Cloudlet	Cloud
State	Only soft state	Hard and soft state
Management	Self-managed; little to no professional attention	Professionally administered, 24x7 operator
Environment	”Datacenter in a box” at business premises	Machine room with power conditioning and cooling
Ownership	Decentralized ownership by local business	Centralized ownership by Amazon, Yahoo!, etc.
Network	LAN latency/bandwidth	Internet latency/bandwidth
Sharing	Few users at a time	100s-1000s of users at a time

2.3. How Cloudlet Works?

The cloudlet architecture can be explained as a high level architecture for code offloading in a hostile environment. Cloudlets are the middle element in a three-tier architecture i.e. intermediate layer between the cloud infrastructure and the mobile device Figure 3.

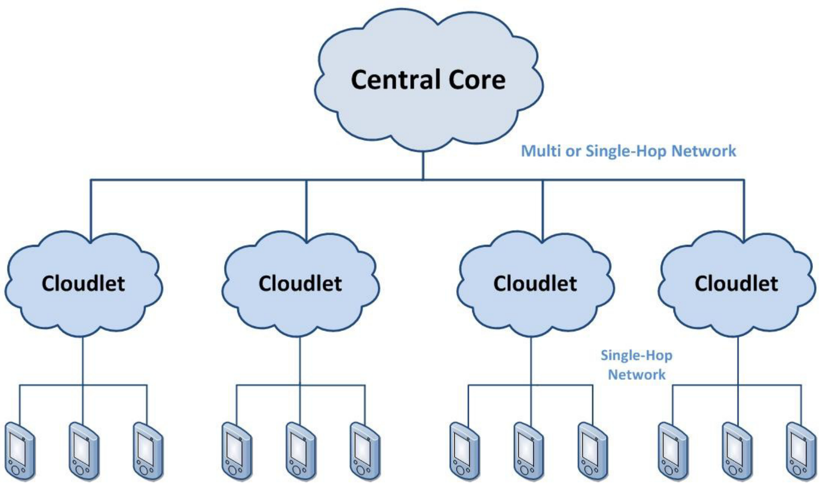

Fig. 3. Three-Tier Architecture: Mobile Device, Cloudlet, Central Core (Cloud).

The heart of this architecture is a large central core, which can be implemented as one of the Amazon's, Microsoft's, Google's data centres or private enterprise clouds. The other end of the architecture includes offload elements for mobile devices cloudlets. They are located close to the mobile devices they serve. This architecture decreases latency by using a single-hop network and potentially lowers the battery consumption by using Wi-Fi or short-range radio instead of broadband wireless, which typically consumes more energy [6].

3. Specific features of Cloudlets

In this section we present several specific features of cloudlets, such as using cloudlets with thin and fat clients, or interconnection with VMs.

3.1. Cloudlets as offload elements

A key attribute in this architecture is that offload elements are stateless [7]. A mobile device does not need to communicate with the core elements during offloading; they need to communicate with the closest cloudlet. Communication with the central core elements is done only during provisioning and configuration.

One approach to offload is the VM synthesis. In this case an application overlay is offloaded from the mobile device to the cloudlet. VM synthesis is useful in hostile environment characterized by unreliable networks, loss of cyber foraging platforms and a need for rapid deployment. For example, we can take a scenario where the first responder must execute a computation intensive application configured to work with cloudlets. At runtime, application discovers nearby cloudlet and offloads the computation intensive mobile application [8]. However, due to loss of network connectivity, limited energy sources on the cloudlet, the mobile application disconnects from the cloudlet. The mobile application can locate a different cloudlet and run the application in a short time, with no need of configuration on the application or the cloudlet. This flexible approach enables the use of available resources, replace disconnected resources and dynamically customize newly resources.

The following features are essential benefit for offloading.

They:

- are discoverable, generic, stateless servers located in single-hop proximity of mobile devices;
- run a separate virtual machine (VM) for each offloaded application;
- enhance processing capacity and conserve battery power while at the same time providing ease of deployment in the field; and
- communicate with the central core only for provisioning.

3.2. Using cloudlets with thin/fat clients

In the previous sections we explained that cloudlets are used for processing of resource rich applications. In the last few years we are witnessing the evolution of mobile devices and their enhanced performances (CPU processing, memory, battery life, etc.). Because the way of processing the application is quite a challenge, is it better to use thin mobile clients and using clouds (in case cloudlets) for processing or to use fat clients which will make most of the processing on the device by itself.

3.2.1. Thin Clients: Applications which are used for face recognition, voice to text translation, etc. cannot be processed using only mobile devices because there are huge algorithms which have to be processed in background, huge database, resources which cannot be placed on the phone. Some of these applications are FaceLock [9] for facial recognition, Siri [10] (designed/developed by Apple), which enables the customer to use their voice in order to send messages, schedule a meeting, make a phone call, etc. The Processing of these applications can be done by using afore mentioned cloudlets. Cloudlets could be used for fast processing of the tasks given from the mobile device.

3.2.2. Fat Clients: Fat clients are mobile devices with performances to process resource rich applications. The processing is done on the client side, and not on the server. Instagram [11], one of the most popular applications in the moment, works this way. With Instagram [11], a customer can take a photo or video, choose filter to transform its look, feel and then post and share on the social networks. The transformation of the photo and video using this application is done directly on the mobile device; only a small part of the processing is done on the cloud. In this scenario the usage of cloudlets would be really small.

In between these extremes there are many possible ways to partition an application based on application characteristics, data content, etc. There is a wide range of tools and techniques to help with this partitioning. Some require operating system support while others would need lower level system support.

Process models may also vary widely, from a single process to a collection of processes. At any given time many mobile devices can use the cloudlet. Sometimes one mobile client can offload multiple concurrent applications on same cloudlet. The offload support for some applications may require a Linux environment, while others require specific Windows environment. So the ability to support offload via a wide range of operating systems, programming languages, and process models is essential. In addition, a good isolation between offloaded executions is advisable for safety and robustness [8].

3.3. Use of virtual machines in Cloudlet

The usage of VM in cloudlets enables clean separation. The complex problem of configuring software on the cloudlet to service mobile devices is avoided. Instead, the problem is transformed into a simpler problem of rapidly delivering a precisely preconfigured VM to the cloudlet. A VM cleanly encapsulates and separates a transient guest software environment from the permanent host software environment of the cloudlet infrastructure [8]. The interface between host and guest is stable and narrow. This ensures longevity of the cloudlet and increases the chances of compatibility between cloudlet and mobile device.

The VM approach is weaker than alternatives such as software virtualization or process migration. It is more general than language based virtualization approaches which require applications to be written in specific languages such as Java or C#. One VM image is many gigabytes in a size. In a hostile environment, efficient dissemination of VM to the cloudlet is a major challenge. When a cloudlet acquires a copy of VM, it can treat it as a persistent cache copy and keep it until the space has to be reclaimed [12]. Mobile devices can be connected again in the future with the same cloudlet, with persistent caching of VM's. Also, the same VM can be used for offloading by many mobile devices at the same time.

4. Discussion

Most of the papers released in the last 2 years cover the cloudlet architecture and the way of how it is working. The main focus is on the applications for image and voice recognition, natural language processing, mission planning, decision making, etc. Authors experiment how offloading works by using cloudlets and how they work with offloading directly on the cloud. In these experiments the main goal is to check the performance of the cloudlet concept, processing time of the application, time of response of the cloudlet, time of distribution of the information to the cloudlet, etc.

The most attractive research area is face recognition. Numerous techniques have been designed to detect and recognize faces. Almost all of the experiments in which offloading to the cloudlet is conducted are connected with face recognition.

In last few years there is a battle about the devices that should be used for offloading to the cloud. There are many mobile device types used by the customers nowadays. These mobile devices can operate as thin or fat clients. Thin clients have no custom application code and completely rely on the server for their functionality. Fat clients typically have one to three layers of application code on them and can operate independently from a server for some period of time. Typically, fat clients are most useful in situations where communication between a client and server cannot be guaranteed. Fat clients depend heavily on the operating system and mobile device type and the code can be difficult to release and distribute.

4.1. Proof of Cloudlet concept

Still there are open research questions and various research hypotheses. The first is the proof of the cloudlet concept. For example, the hypothesis is whether the use of a cloudlet gives better performance when cloud offloading has been made, instead of directly processing on the cloud without using the middle architecture stack layer. We expect that there are specific domains where this concept as middleware offers better performance and service than direct communication to the cloud. In this case, the open research problem is to determine this domain.

The on-going research for proof of the cloudlet concept includes a benchmark application that offloads complex computations on the cloudlet and indirectly onto clouds compared to offloading directly to the cloud, without using a cloudlet. Usual approach in this case is to check the performance of the application execution in both ways and compare the results. There is another way to check the performance of the application, by using only the mobile device. Although mobile devices today are very powerful, the benchmark applications, which we are discussing are using resources beyond the capabilities of current technology mobile's device resources.

4.2. Balance of offload and mobile device processing

Another research topic tackles the open problem to choose whether it is better to use thin or fat mobile devices. This would give an answer to the long-standing debate about whether it is better to carry out processing directly on the mobile device and it would have spent the same resources or it would be better if the processing was performed on an external resource and it would keep the resources of the device free for other purposes.

The number of the mobile devices with high performances is still small. There are resource-rich developed applications, which need to offload to the cloud in order to use his resources for processing which makes this approach not possible. At least, the ongoing research would determine the domain when the thin client approach is better than fat client in mobile devices, i.e. will determine the balance of offload elements and host mobile device processing.

This can be verified by extensive experiments on various benchmark applications, although it is platform dependent and we are aware that new mobile devices with increased power and capabilities are announced at least once per year. Also the problem requirements are on a rise, for example, in face recognition, the database is enormously increasing every day. We expect that future applications will use a kind of balancer that will determine what can be processed locally in real-time, what can be offloaded to a cloud using Internet supported by mobile operator and what can be offloaded on a cloudlet via Wi-Fi connection.

One of the challenges will be to successfully offload the benchmark application to the nearest cloudlet and try to configure cloudlet processing and communication capabilities. It will enable conclusions on how to use cloudlet resources as much as possible and release the mobile device resources for other purposes.

5. Conclusion and Future Work

From the presented above, as well as from the number of studies that have been made on this topic, we can conclude that the use of cloudlets is of great benefit. They allow faster data transfer, faster processing of applications, smaller use of mobile device resources, etc. Without their presence, the processing is carried out directly on a distant cloud, and thus the communication and processing time is longer, the use of memory on the device is higher, battery life of the device is smaller, which is definitely something that would annoy the mobile device user. Taking this into consideration, we can conclude that using cloudlets would be of great benefit for mobile services customers, Especially if they want to use an application that requires resources that the customer's mobile device doesn't have at the moment.

We have presented the architecture of the cloudlet concept and explained basic approaches and features. The main discussion of cloudlet challenges and research open questions included:

- proof of cloudlet concept,
- balance of offload elements and host mobile device processing, and
- balance of offload elements via Wi-Fi on cloudlet and mobile operator Internet communicated cloud.

The following are hot research topics:

- Determine domains where the cloudlet concept with middleware offers better performance for offloading instead of direct communication to clouds
- Determine domains where the local mobile device processing is better than using offload elements
- Determine balance of local mobile device processing and offload elements

Although most of this research is experimental, it includes development of new protocols and architectures in cloud environment. Specific experimental setup includes establishing a real organization and implementation of a cloudlet and clouds, than benchmark applications and various mobile device platforms. We expect that the overall dilemma about using cloudlet as an intermediate layer to offload to the cloud, or using offloading directly to the cloud in order to enable more efficient and cheaper solution will result in determination of specific domains where each configuration impacts the most. With this research maybe we can change the course of offload computing to the cloud, application development, use of VM and cloudlets.

References

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, 2009
[2] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer, vol. 43, no. 4, pp. 51–56 2010
[3] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," ACM SIGPLAN Notices, vol. 47, no. 4, pp. 171–184, 2012
[4] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in Proceedings of the third ACM workshop on Mobile cloud computing and services. ACM, 2012, pp. 29–36
[5] S. Simanta, K. Ha, G. Lewis, E. Morris, and M. Satyanarayanan, "A reference architecture for mobile code offload in hostile environments,"in Mobile Computing, Applications, and Services. Springer, 2013, pp. 274–293
[6] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 49–62
[7] M. Satyanarayanan, "From the editor in chief: Augmenting cognition," Pervasive Computing, IEEE, vol. 3, no. 2, pp. 4–5, 2004
[8] G. Lewis, M. Satyanarayanan, S. Simanta, and K. Ha, "Cloud offload in hostile environments," 2011
[9] FaceLock. (2013, Aug.) Facelock. [Online]. Available: https://play.google.com/store/apps/details?id=com.facelock4apps&hl=en
[10] Apple. (2013, Aug.) Siri. [Online]. Available: http://www.apple.com/ios/siri/
[11] Instagram. (2013, Aug.) Instagram. [Online]. Available: http://instagram.com/
[12] H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. De Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in Proceedings of the 4th ACM European conference on Computer systems. ACM, 2009, pp. 1–1