JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 11 : 2671~2676 Nov. 2015

# 모바일 장치와 구름을 사이에 거리의 효과 분석설계

니욘사바에릭[1] · 장종욱[2*]

# A Design of Analyzing effects of Distance between a mobile device and Cloudlet

**Niyonsaba Eric[1] · Jong-Wook Jang[2*]**

[1]Department of Computer Engineering, Dong-Eui University, Busan, Korea
[2*]Department of Computer Engineering, Dong-Eui University, Busan, Korea

## 요 약

오늘날, 모바일 장치는 현재 응용 프로그램의 넓은 범위를 지원 할 수 있습니다. 불행히도, 일부 응용 프로그램은 늘어나는 계산 능력을 필요로 하고 모바일 장치는 이러한 낮은 처리 능력, 제한된 메모리, 예측할 수없는 연결 및 제한된 배터리 수명으로 제약에 자원이 한정되어 있습니다. 모바일 장치의 한계를 극복하기 위해 연구자들은 강력한 컴퓨팅 인프라 모바일 장치에서의 작업 부하를 이동하기 위해 가상화 기술을 사용하여 모바일 장치에 클라우드 컴퓨팅 서비스를 확장하는 상상합니다. 이러한 기술은 인프라 자원이 풍부한 클라우드 또는 서버에 모바일 장치에서 자원을 많이 소비하는 계산의 이행으로 구성되어 있습니다. 본 논문에서 클라우드 아키텍처 기능에 클라우드과 모바일 장치 사이의 거리의 영향을 분석합니다.

## ABSTRACT

Nowadays, Mobile devices are now capable of supporting a wide range of applications. Unfortunately, some of applications demand an ever increasing computational power and mobile devices have limited resources due to their constraints, such as low processing power, limited memory, unpredictable connectivity, and limited battery life. To deal with mobile devices' constraints, researchers envision extending cloud computing services to mobile devices using virtualization techniques to shift the workload from mobile devices to a powerful computational infrastructure. Those techniques consist of migrating resource-intensive computations from a mobile device to the resource-rich cloud, or server (called nearby infrastructure). In this paper, we want to highlight on cloudlet architecture (nearby infrastructure with mobile devices), its functioning and in our future work, analyze effects of distance between cloudlet and mobile devices.

**키워드** : 모바일클라우드, 클라우드렛, 모바일 응용 프로그램 모델, 오프로딩 컴퓨테이션

**Key word** : mobile cloud computing, cloudlet, Offloading computation, Mobile application model

**Received** 02 October 2015, **Revised** 30 October 2015, **Accepted** 09 November 2015

*** Corresponding Author** Jong-Wook Jang (E-mail:jwjang@deu.ac.kr, Tel:+82-51-890-1709)
Department of Computer Engineering, Dong-Eui University, Busan , Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.11.2671 print ISSN: 2234-4772 online ISSN: 2288-4165

## Ⅰ. INTRODUCTION

Cloud computing and mobile applications have been the leading technology trends in recent years. Mobiles devices such as smartphones and tablets have become an essential part of our lives, because of their powerful capabilities. Users depend on their mobile devices to make calls, create and edit documents, perform image processing, access the online social networks websites (Facebook, twitter, etc.), organize meetings and make video and audio calls. In spite of the benefits provided by the mobile devices, and the way they make the life easier, they have many weaknesses such as limited battery life-time, limited processing capabilities and limited storage capacity. It is very important to take into account these limitations, since they are hindering mobile users from doing their daily tasks in an efficient way. One solution to overcome these limitations is to integrate Cloud Computing technology with mobile devices to produce what is called Mobile Cloud Computing (MCC) [1].

In MCC, the processing and storage of intensive jobs are transferred to the resources rich cloud system to take place there. In case of jobs that need high processing and computing capabilities, these jobs are migrated to the cloud where intensive processing can be done, and thus the final result is returned back to the mobile device. With this technique, Cloud Computing resolves both problems: limited processing capabilities and limited power of the mobile phones. On the other hand, if there are files, videos and images with a very large size, they will be transferred to the storage inside the cloud; and whenever the mobile user needs any of them, one can request it from the cloud. With this technique CC resolves the problem of limited storage capacity of the mobile phones. Rather than running applications locally and directly requesting data from content providers, a mobile device can offload parts of their workload to the cloud, taking advantage of the abundant cloud resources to help gather, store, and process data for the mobile device [2].

MCC has recently become one of the most important and hottest research topics; because it integrates mobile devices with the cloud computing technologies. In general, to make the smartphones energy efficient and computationally capable, major hardware and software level changes are needed, which requires the developers and manufacturers to work together [3].

Due to size-constraints, hardware level changes alone may not enable smartphones to achieve true unlimited computational power. Therefore, software-level changes are more effective, where computation is performed on remote resources with partial support of a smartphone's hardware [4]. Moreover, despite the fast development of hardware technology, it is still difficult to support computation-intensive applications (e.g., image processing, augmented reality) on mobile devices, hindering developers from bringing richer experiences and complex applications to mobile users [2]. Therefore, smartphones require an application model that supports computation offloading and optimized for mobile cloud environment in terms of heterogeneity, context awareness, application partitioning overhead, network data cost, bandwidth, and energy consumption [5].

## Ⅱ. OVERVIEW OF APPLICATION MODELS FOR MOBILE CLOUD COMPUTING

The mobile cloud application models are designed to achieve a particular objective, such as executing applications that have insufficient resources for local execution, enhancing applications performance (in terms of computation time), or achieving energy efficiency on mobile devices[5]. Some of the proposed applications models have been discussed in literature review such as:

1) CloneCloud: CloneCloud [6, 7] is a system which automatically convert applications of the mobile devices by partially offload it into the virtual clone (phones) present in the cloud. The synchronization of the smartphone and its clone is very important for consistent execution.

Therefore, when augmentation is required, the smartphone application process enters a sleep state and transfers the process state to the clone. The VM creates a new process state and overlays the received information, followed by execution of the clone. On completion of the execution, the process state of the clones' application is sent to the smartphone, where the process state is reintegrated into the smartphones' application and the application comes out of a sleep state. 2) WebLet [8]: This model is based on elastic applications technique, where a single elastic application is partitioned into multiple components called weblets.

A weblet can be defined as an independent functional unit of an application that can compute, store, and communicate while keeping its execution location transparent. The topology of the elastic applications falls into multiple types of patterns, called elasticity patterns. Weblets support three types of elasticity patterns, i.e., replication, splitter and aggregation. 3) ThinkAir: ThinkAir [9] supports method-level offloading to a smartphone clone executing in the cloud. It is designed to achieve the desired QoS (quality of service) by executing multiple clones of the smartphone in parallel. ThinkAir requires minor modifications in the source code of the applications.

Therefore, it is the duty of the programmers to identify all resource intensive methods that can be offloaded to the cloud for remote execution. When a remoteable (offloadable) method is called, ThinkAir starts the profilers to monitor the remoteable method and store the information for future offloading decisions. Moreover, the execution controller makes a decision about the execution location of the method that is based on the execution time, energy consumption, and previous execution history kept by the profilers. There are also other application models such as Cloudlet, Excloud, MAUI, and Cuckoo….In this paper, we will highlight to Cloudlet model in details.

## Ⅲ. CLOUDLET APPLICATION MODEL

A cloudlet is a new architectural element that arises from the convergence of mobile computing and cloud computing. It represents the middle tier of a 3-tier hierarchy: mobile device - cloudlet -cloud [10]. A cloudlet can be viewed as a "data center in a box" whose goal is to "bring the cloud closer". A cloudlet has four key attributes: 1.only soft state: It is does not have any hard state, but may contain cached state from the cloud. 2. Powerful, well-connected and safe: It possesses sufficient compute power (i.e., CPU, RAM, etc.) to offload resource-intensive computations from one or more mobile devices. It has excellent connectivity to the cloud (typically a wired Internet connection) and is not limited by finite battery life (i.e., it is plugged into a power outlet). 3. Close at hand: It is logically proximate to the associated mobile devices. "Logical proximity" is defined as low end-to-end latency and high bandwidth (e.g., one-hop Wi-Fi). Often, logical proximity implies physical proximity. However, because of "last mile" effects, the inverse may not be true: physical proximity may not imply logical proximity.4.builds on standard cloud technology: It encapsulates offload code from mobile devices in virtual machines (VMs), and thus resembles classic cloud infrastructure such as Amazon EC2 and OpenStack.

### A. Cloudlet: Physical Architecture


**Fig. 1** Cloudlet concept [11]

Rather than relying on a distant "cloud," the resource poverty of a mobile device can be addressed by using a nearby resource-rich "cloudlet". The need for real-time interactive response can be met by low latency, one-hop, and high-bandwidth wireless access to the cloudlet [12]. The mobile device functions as a thin client, with all significant computation occurring in the nearby cloudlet. Physical proximity of the cloudlet is essential: the end-to-end response time of applications executing in the cloudlet needs to be fast (few milliseconds) and predictable. If no cloudlet is available nearby, the mobile device can gracefully degrade to a fallback mode that involves a distant cloud or, in the worst case, solely its own resources. Full functionality and performance can return later, when a nearby cloudlet is discovered. As Figure 1 illustrates, cloudlets are decentralized and widely-dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile computers. A cloudlet can be viewed as a "data center in a box." It is self-managing, requiring little more than power, Internet connectivity, and access control for setup. Internally, a cloudlet may be viewed as a cluster of multi-core computers, with gigabit internal connectivity and a high-bandwidth wireless LAN. For safe deployment in unmonitored areas, the cloudlet may be packaged in a tamper-resistant or tamper-evident enclosure with third-party remote monitoring of hardware integrity. If a mobile device user moves away from the cloudlet he is currently using, interactive response will degrade as the logical network distance increases. To address this effect of mobility, the offloaded services on the first cloudlet need to be transferred to the second cloudlet maintaining end-to-end network quality.

### B. Computation Offloading Cloudlet System

Cloudlet is discoverable by mobile devices and is virtual-machine based to promote flexibility, mobility, scalability and elasticity. The mobile devices act like a thin client, and offload resource-intensive tasks to the cloudlet.

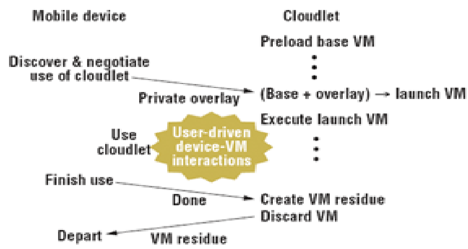


**Fig. 2** Dynamic VM Synthesis Timeline Approach[13]

Moreover, the mobile devices rely on low-latency, one-hop cloudlet that is accessible via a Wi-Fi connection. For this approach (Dynamic VM Synthesis), a small VM overlay is delivered by a mobile device to cloudlet infrastructure that already possesses the base VM from which this overlay was derived. The infrastructure applies the overlay to the base to derive the launch VM, which starts execution in the precise state from which the overlay was derived. Figure 2 illustrates the steps of this approach. Cloudlet stands for low latency, processing power and high bandwidth.

## Ⅳ. WORK DESIGN

Our design will be done in two steps namely implementation of Cloudlet infrastructure and programming on android platform in order to measure the impact of distance between cloudlet and associated mobile devices.


**Fig. 3** Work design Architecture

Cloudlet and mobile devices must be close but there no idea in which range. In our experiments, we will consider different distances between cloudlet and mobile device in order to make a conclusion of an acceptable range of distances to benefit highly cloudlet's power.

## Ⅴ. CONCLUSION

Mobile devices have very limited resources, being their main weak points computing power, storage space, and battery life. To augment computing power and improve battery life, many of mobile applications models have developed so far by some researchers. Among them, Cloudlet which will be used in our design have been detailed in order to use it in our implementation. Our following work will be the implementation of our system design on which our experiments will rely.

### ACKNOWLEDGMENTS

This work was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2015H1C1A1035898).

## REFERENCES

[ 1 ] Yaser Jararweh , Lo'ai Tawalbeh , Fadi Ababneh , Abdallah Khreishah , Fahd Dosari; Scalable Cloudlet-based Mobile Computing Model; *The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014), Procedia Computer Science 34*(2014) 434-441.

[ 2 ] Fangming liu,Peng shu,Hai jin,Linjie ding,and Jie yu, Huazhong,Di Niu and Bo li; Gearing Resource-poor mobile devices with powerful clouds: *Architectures, Challenges, and Applications in Wireless Communications IEE*E (Volume :20,no:3) p.14-22, 2013.

[ 3 ] N. Vallina-Rodriguez and J. Crowcroft, “Energy management techniques in modern mobile handsets,” *IEEE Commun. Surveys & Tutorial*s, vol. 99, pp. 1-20, 2012.

[ 4 ] R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, and H. Bal, “eyedentify: Multimedia cyber foraging from a smartphone,” in Multimedia, 2009. *ISM’09. 11th IEEE International Symposium on. IEEE*, 2009, pp. 392-399.

[ 5 ] Atta ur Rehman Khan, Mazliza Othman, Sajjad Ahmad Madani, IEEE Member, and Samee Ullah Khan; A Survey of Mobile Cloud Computing Application Models; *IEEE communications surveys & tutorial*s, vol. 16, no. 1, first quarter 2014 p.393-413.

[ 6 ] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in *Proceedings of the sixth conference on Computer systems, ser. EuroSys ’11*. New York, NY, USA: ACM, 2011, pp. 301-314.

[ 7 ] B.-G. Chun, S. Ihm, P. Maniatis, and M. Naik, “Clonecloud: boosting mobile device applications through cloud clone execution,” *arXiv preprint arXiv*: 1009. 3088V2, 2010.

[ 8 ] X. Zhang, S. Jeong, A. Kunjithapatham, and S. Gibbs, “Towards an elastic application model for augmenting computing capabilities of mobile platforms,” in *Mobile wireless middleware, operating systems, and applications*. Springer, 2010, pp. 161-174.

[ 9 ] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Unleashing the power of mobile cloud computing using thinkair,” *arXiv preprint arXiv*: 1105.3232, 2011.

[10] (Online) http://Elijah.cs.edu

[11] (Online) http://www.google.co.kr/search?q=Cloudlet

[12] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, Nigel Davies; the Case for VM-based Cloudlets in Mobile Computing, *Pervasive Computing,IEE*E Vol 8,no.4 p.14-23.

[13] Grace A. Lewis, Sebastian Echeverría, Soumya Simanta, Ben Bradshaw, James Root at Carnegie Mellon Software Engineering Institute 4500 Fifth ave., Pittsburgh, PA USA ; *Cloudlet-Based Cyber-Foraging for Mobile Systems in Resource-Constrained Edge Environments*.

니욘사바에릭(Niyonsaba Eric)

2015년 ~ 현재 : 동의대학교 컴퓨터공학과 석사
※관심분야 : 모바일 네트워크

장종욱(Jong-wook Jang)

1995년 2월 부산대학교 컴퓨터공학과 박사
1987년 ~ 1995년 ETRI
2000년 2월 UMKC Post-Doc.
1995년 ~ 현재 동의대학교 컴퓨터공학과 교수
※관심분야 : 유무선통신시스템, 자동차네트워크