Benchmark Applications Used in Mobile Cloud Computing: A Systematic Mapping Study

Francisco Airton Silva, Paulo Maciel, Eder Quesado, Germano Zaicaner, Matheus Dornelas, Bruno Silva
Federal University of Pernambuco (UFPE) – Recife, Pernambuco, Brazil
{faps,prmm,eraqs,gz,mdr,bs}@cin.ufpe.br

***Abstract*—Mobile Cloud Computing (MCC) integrates mobile computing and cloud computing aiming to extend mobile devices capabilities through offloading techniques. In MCC, many controlled experiments have been performed using mobile applications as benchmarks. Usually, these applications are used to validate proposed algorithms, architectures or frameworks. The task of choosing a specific benchmark to evaluate MCC proposals is difficult because there is no standard applications list. This paper presents a systematic mapping study for benchmarks used in MCC research. Taking five months of work, we have read 763 papers from MCC field. We catalogued the applications and characterizes them considering three facets: category (e.g.: games, imaging tools); evaluated resource (e.g.: time, energy); and platform (e.g.: Android, iPhone). The mapping study evidences research gaps and research trends. Providing a list of downloadable standardized benchmarks, this work can aid better choices to guide more reliable research studies since the same application could be used for different scientific purposes.**

***Index Terms*—Mobile Cloud Computing; Offloading; Partitioning; Performance Evaluation**

I. Introduction

Cloud computing can be defined as the aggregation of computing as a utility and software as a service where applications are delivered as services over the Internet and data centers provide those services [1]. In another side there are smartphone applications increase in complexity and required resources. Unfortunately, the advances in smartphone hardware and battery life have been slow to respond to the computational demands of applications evolved over the years. Hence, many applications are still unsuitable for smartphones due to constraints, such as low processing power, limited memory, unpredictable network connectivity, and limited battery life [52].

To tackle this problem a strategy called offloading is applied. Computation offloading is a process that migrates resource-intensive computations from a mobile device to the resource-rich cloud (called cloudlet, in case of nearby infrastructure). Cloud based computation offloading enhances the applications performance, reduces battery power consumption, and execute applications that are unable to execute due to insufficient smartphone resources.

Mobile cloud computing (MCC) is an integration of cloud computing technology with mobile computing in order to make mobile devices resource-full in terms of computational power, memory, storage, energy, and context awareness. A significant amount of research has been performed on computation offloading in such a field [18], [9], [33], [53]. These studies usually focus on: why to offload (improve performance or save energy); when to decide offloading; what mobile systems use offloading; and which are the infrastructures for offloading. Aiming to conduct these studies, most of the researchers adopt mobile applications to prove their hypotheses, when proposing new theories. However, there is no place with a list of possible applications that could be used in experiments in mobile cloud computing. Consequently, researchers might not know the level of adoption of a specific application in the field. They may also be unaware of which platform (e.g.: iPhone) is more used combined with a specific application category (e.g.: Mathematical app). These pieces of information are useful to guide new research studies and standardize the characteristics of controlled experiments with offloading techniques. This paper provides such information by addressing and answering the question "What are the benchmark offloaded applications used in MCC and which characteristics define them?".

This paper presents a systematic mapping study, performed in order to map out the applications used in MCC field. By means of analyzing three facets (category, platform, and evaluated resource), we synthesize implications for practicing, identifying research trends, open issues, and areas for improvement. A mapping study is an evidence-based approach, applied in order to provide an overview of a research area, and identify the quantity and type of research.

The remainder of this paper is organized as follows: In Section II, the systematic mapping study method is better described; Section III reports the findings based on the frequency of applications use; Section IV presents the related work; Section V draws some conclusions and provides recommendations for further research on this topic.

II. Systematic Mapping Study Process

A mapping study is a systematic process that provides an overview and summarizes published paper results of a particular research area, by answering questions and categorizing studies. As main benefit, it can be used to identify gaps in the existing research that will lead to topics for further investigation. Thus, a systematic mapping study was used in this research to "map out" the benchmarks used in mobile cloud computing. The study follows the systematic mapping process proposed by Petersen et al. [43]. The essential process, is composed of five steps with specific outcomes and each phase is discussed in the following sections: (*Definition of*

Research Questions, *Conducting Search*, *Screening of Papers*, *Keywording* and *Data Extraction*).

The main reasons to perform a mapping study, and consequently the contributions of this paper, can be stated as follows [2]:

- To make an unbiased assessment of as many studies as possible, by identifying existing gaps in current research and contributing to the research community with a reliable synthesis of the data. The gaps point out clear opportunities of research topics. For instance, in MCC, which mobile phone platform is less explored, BlackBerry or iPhone?;
- To provide a systematic procedure for identifying the nature and extent of the empirical study data that is available to answer research questions. By following a systematic procedure, the research can be replicated in future, updating the results. In our study, we list the most used applications in MCC. This list can be improved along with the time;
- To assist planning new research, avoiding unnecessary duplication of effort and error. Since we catalogue the benchmark applications used in MCC, probably it is not interesting for researchers to adopt an application outside such a list;

A. Conduct Search

The second step is to prepare search strings to use in different digital libraries and then collect related papers. Firstly, an automatic search was conducted in different search engines (IEEEXplore, ACM Digital Library, Scopus, ScienceDirect and Springerlink). It is important to mention that all search strings were calibrated regarding each search engine. In total, 763 studies were collected by applying that search strategy.

B. Screening of Papers

In the screening phase, the papers initially collected are filtered to remain only papers that may answer the research question. In this work, we applied three filters. Starting with 763 papers, the first filter was applied focusing on the title of the identified studies, resulting in 491 studies. The large number of duplicated studies contributed to this large difference. The second filter was applied on the title and abstract, which resulted in 113 studies. The last filter focused on the introduction and the evaluation section, resulting on 47 studies that we have read as a whole: [42], [3], [54], [32], [53], [20], [61], [56], [49], [5], [39], [26], [16], [48], [57], [22], [34], [47], [29], [50], [60], [27], [36], [14], [11], [6], [8], [23], [25], [7], [13], [19], [46], [12], [58], [35], [44], [10], [59], [31], [63], [62], [15], [65], [55], [17], [45].

C. Keywording

Researchers must identify possible facets to characterize the papers and derive useful information. A classification scheme is a mechanism composed of a categories set used to classify the primary studies such a way it extracts detailed information and identifies research gaps. Aiming to build our classification scheme, we based it on a systematic process proposed by [43] called keywording. Three different facets were used derived from the studies found, namely following:

1) **Category**: Mathematical tools, Games, Standalone, Web Applications, Video Streaming, Text Search, Antivirus and Imaging Tools
2) **Evaluated Resource**: Time, Energy and CPU/Memory Usage.
3) **Platform**: Android, Windows, iPhone, Android x86, Maemo 5 Linux Blackberry and Simulation (using only computers without devices emulation).

D. Data Extraction

A data extraction form was designed in order to gather the required information to address the objectives of this study, classifying and answering the research question. The full paper was read and the following information was extracted from each study: the research facets (category of each application, evaluated resource, and platform), the application's name and the URL to download it (whenever it was available).

To organize this information, spreadsheets were adopted to document the data extraction process. They contained each category of the classification scheme and the position inside the paper which answered the research question. All data and classification can be found in the web-page: *http://cin.ufpe.br/˜faps/mapping*.

III. Mapping by Quantities of Occurrence

This section presents our findings, highlighting evidences gathered from the data extraction process along with comments about the results. In this section, the results are presented according to the quantity of studies classified by each facet.

A. Application Category

The facet Category classifies the studies according to their main functionality or utility. Figure 1 presents the quantity of papers related to its respective category. It can be observed that the vast majority of applications used as benchmarks are categorized as Imaging Tools. These applications are tools that somehow manipulate images and usually requires a high amount of resource [55]. It is a hard task to trace conclusions about reasons to the rankings presented in this mapping study. However, mobile cloud experiments usually test strategies to decrease the time and save energy, so it is assumed that applications that require more resources are more susceptible to be adopted.

Mathematical Tools also have great demand for resources as seen in examples like sorting algorithms, Fibonacci, etc. They are easy to use once the entrances are usually very simple numerical data with high scalability that makes the experiments flexible. **Games** are becoming more and more popular not only for PC platforms but also mobile devices and they present innovations like 3D graphics and multiplayer modes. In mobile field developers deal with scarcity of resources as the major challenge [24]. **Web applications** on mobile devices are already a necessity, as users want to consume information

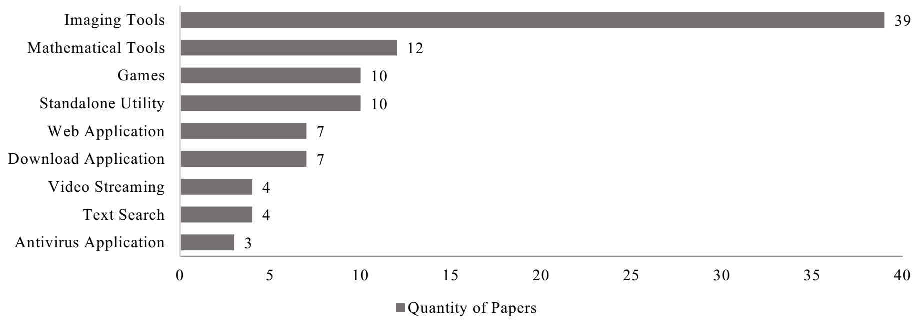

Fig. 1: Application Category.

and interact with other people. Hence, the need to access some web pages is enhanced, such as Google, Facebook, Youtube, etc. There are mobile application versions for these web pages that can provide useful content to users, taking advantage of mobile nature features such as GPS location service [40].

The easy access of Wi-Fi internet contributed to **Video Streaming** being highly used through mobile devices. Namboodiri et al. [41] tested three applications that belong to the categories highlighted in this article: text search (word processing), multimedia applications and games. They aimed to verify when it is advantageous to carry out the execution of these applications on remote servers in terms of energy consumption. Results showed that running applications in cloud consumes more battery than when running locally, due to Wi-Fi connection. The increased adoption of **Anti-virus** for mobile platforms is related to the rapid malware emergence, especially for Android operating system. According to Yajin et. al [64], between 2010 and 2012 there was an increase of 400 percent in amount of malware for Android. Majority of malwares come from installing applications from PlayStore.

B. Imaging Tools

Due to the expressive quantity of papers using Imaging Tools, (illustrated in Section III-A) we investigated more closely this topic in order to know what type of applications researchers are using more. Figure 2 illustrates that Face Detection and Face Recognition had greater adoption. It might be justified by the fact that nowadays various commercial applications provide detection functionality, whereas recognition is very useful to security contexts [51].

.

C. Evaluated Resource

With this facet, we desire to identify the investigated metrics used as dependent variables. Figure 3 shows that the most evaluated resources are **Time and Energy**.

Energy consumption is so used as metric of quality in mobile cloud research because it has always been a users' concern and consequently a mobile industry interest. Many applications are still unsuitable for smartphones due to hardware constraints [28]. Computing speeds of these mobile devices, however, will not grow at the same pace as servers' performance. This is due to several constraints, including: Form Factor, as users want devices that are smaller and thinner and yet with more computational capability; Power Consumption, insofar the current battery technology constrains the clock speed of processors, doubling the clock speed approximately octuples the power consumption. Consequently, it is difficult to offer long battery lifetimes with high clock speeds [33]. Finally, not so explored, the impact for **CPU** and **Memory** resources depends on the type of application, there are applications that use more CPU than memory and vice versa [62].

Due to the expressive quantity of papers using Time as a dependent variable (illustrated in Section III-C), we investigated more closely this topic to know what type of metric the researchers have been evaluated more. Figure 4 presents three types of metrics related to time. The communication time is the transmission time taken between the mobile device and the cloud. Time to process is the time taken by the task to process inside the cloud. Finally, the elapsed time is the sum of communication time and time to process.

The most common metric to assess mobile applications performance is total execution time or also called elapsed time. Great importance is given to this factor because it influences directly on user experience. When taking into account the elapsed time or communication time, the quality of internet and client-server proximity influences the results. Instead, evaluating the time to process, the number of servers interferes directly in the process runtime.

D. Platform

Platform represents the testbeds used by the experiments in MCC. In most of the cases, platforms are, in fact,the mobile devices used in the experiments. Nonetheless, simulated devices were utilized in conjunction with a computer desktop without any special program. We included the use of Android-X86 which are an unofficial initiative to port Google's Android mobile operating system to run on devices powered by AMD and Intel x86 processors. Figure 5 depicts the results regarding platforms.

The three most popular mobile OS are Android, iOS (iPhone OS) and Windows Phone, and they have similar functionalities. Still, many people speak about security and quality on iOS applications or communication features of Windows

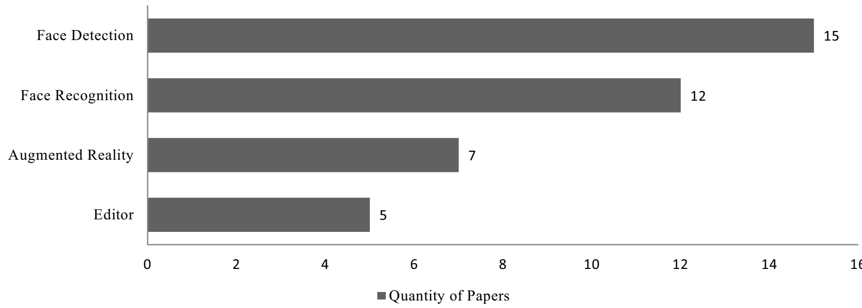

Fig. 2: Imaging Tools Subgroups.

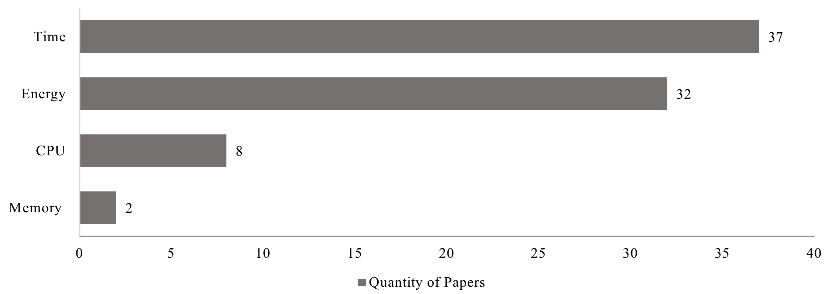

Fig. 3: Evaluated Resource.

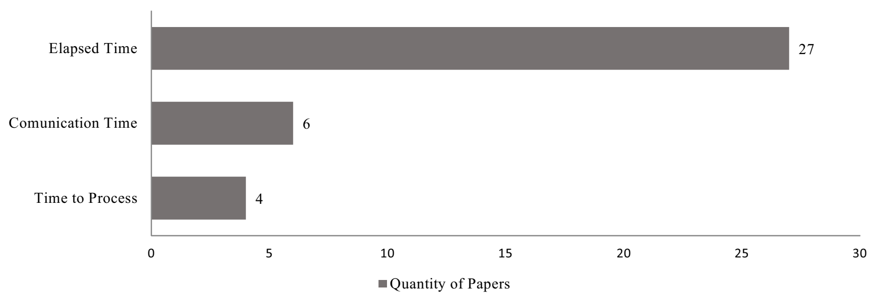

Fig. 4: Evaluated Time Resource.

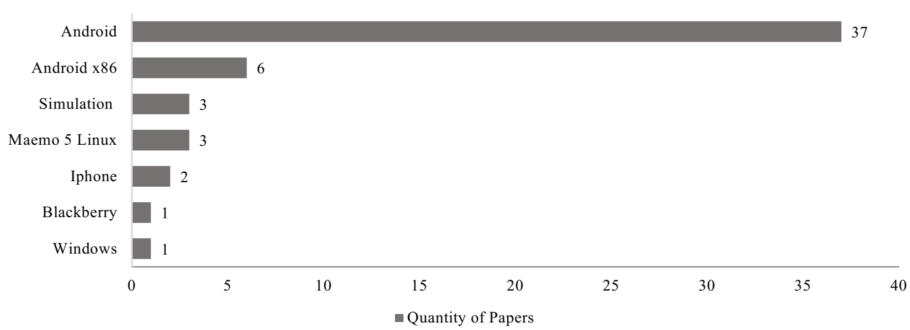

Fig. 5: Platforms.

Phone, but the majority of smartphones around the world have Android running on it [21]. Considering MCC research context, although iPhone and Windows Phone are very popular platforms, they are not positioned at the top of ranking.

In MCC, offloading is the task in which a mobile device sends a workload to be processed by a remote server or virtual machine. A myriad of studies proved that this strategy may save high amounts of time and energy [18], [9]. Android x86 is adopted on many papers and it runs on the virtual machines very easily, without the necessity of source-code adaptation.

IV. Related Work

In mobile cloud computing there are a number of surveys [29], [33], [4], but to the best of our knowledge no one systematic mapping study. Montesi et al. [38] explains that a survey basically answers the question: "What is currently known about this area, and what does it mean to researchers and practitioners?". It should supply the basic knowledge to enable new researchers to enter the area, current researchers to continue developments, and practitioners to apply the results. Systematic mapping studies also aims to identify the state of practice or research on a topic and typically identify research trends [30], however in a systematic way by following a research method. Other aspect is the focus of such surveys, only one study [29] tried to classify the types of application categories (mathematical, games, etc.) but just giving some examples of real applications.

V. Conclusions

This paper introduced the results of a systematic mapping study about benchmark applications used in mobile cloud computing by investigating scientific literature production. In the end, starting from 763 papers, 47 filtered studies used applications as benchmarks. It is important to note that our goal for this systematic mapping study was providing an overview of current literature in mobile cloud focusing on applications; it was outside the scope of this paper to evaluate the quality of the studied papers or explain specific findings. Given the current state of MCC research, we judge that there are few studies with controlled experiments using real applications. In our study, only 47 papers used applications to evaluate their proposals, probably because this field is still relatively recent with the first mobile cloud paper dating from 2009 [37]. In most of the cases, the studies did not provide evidences of how other researchers could access and download the applications used, making it hard to replicate their experiments. From the 47 papers, we listed 25 downloadable applications with their corresponding category and URL. We believe that this mapping study generated state-of-the-art information about the main issues because the studied subject can be understood through the provided answers. In future work, more systematic mappings should be conducted to acquire further experience to aid new experiments.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, Apr. 2010.

[2] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham. Using Mapping Studies in Software Engineering. In *Proc. of PPIG 2008*, pages 195–204. Lancaster University, 2008.

[3] G. Cagalaban, S. Kim, and M. Kim. A mobile device-based virtualization technique for m2m communication in cloud computing security. In T.-h. Kim, A. Stoica, W.-c. Fang, T. Vasilakos, J. Villalba, K. Arnett, M. Khan, and B.-H. Kang, editors, *Computer Applications for Security, Control and System Engineering*, volume 339 of *Communications in Computer and Information Science*, pages 160–167. Springer Berlin Heidelberg, 2012.

[4] R.-S. Chang, J. Gao, V. Gruhn, J. He, G. Roussos, and W.-T. Tsai. Mobile cloud computing research - issues, challenges and needs. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pages 442–453, March 2013.

[5] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: Elastic execution between mobile device and cloud. In *Proc. of the Sixth Conference on Computer Systems*, EuroSys '11, pages 301–314, New York, NY, USA, 2011. ACM.

[6] A. Cidon, T. M. London, S. Katti, C. Kozyrakis, and M. Rosenblum. Mars: Adaptive remote execution for multi-threaded mobile devices. In *Proc. of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*, MobiHeld '11, pages 1:1–1:6, New York, NY, USA, 2011. ACM.

[7] L. O. Colombo-Mendoza, A. Alor, Giner, and R. Valencia-garcía. MobiCloUP!: a PaaS for cloud services-based mobile applications. *Automated Software Engineering*, 21(3):391–437, Feb. 2014.

[8] 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 *Proc. of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 49–62, New York, NY, USA, 2010. ACM.

[9] 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*, 13(18):1587–1611, 2013.

[10] H. Eom, P. S. Juste, R. Figueiredo, O. Tickoo, R. Illikkal, and R. Iyer. SNARF : A Social Networking-inspired Accelerator Remoting Framework. pages 29–34, 2012.

[11] H. Eom, P. S. Juste, R. Figueiredo, O. Tickoo, R. Illikkal, and R. Iyer. Machine Learning-Based Runtime Scheduler for Mobile Offloading Framework. *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pages 17–25, Dec. 2013.

[12] H. Eom, P. S. Juste, R. Figueiredo, O. Tickoo, R. Illikkal, and R. Iyer. OpenCL-Based Remote Offloading Framework for Trusted Mobile Cloud Computing. *2013 International Conference on Parallel and Distributed Systems*, pages 240–248, Dec. 2013.

[13] M. Ferber and T. Rauber. Mobile cloud computing in 3g cellular networks using pipelined tasks. In *Proc. of the First European Conference on Service-Oriented and Cloud Computing*, ESOCC'12, pages 192–199, Berlin, Heidelberg, 2012. Springer-Verlag.

[14] N. Fernando, S. Loke, and W. Rahayu. Honeybee: A programming framework for mobile crowd computing. In K. Zheng, M. Li, and H. Jiang, editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 120 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 224–236. Springer Berlin Heidelberg, 2013.

[15] H. Flores, S. N. Srirama, and C. Paniagua. Towards mobile cloud applications: Offloading resource-intensive tasks to hybrid clouds. *International Journal of Pervasive Computing*, 8(4):344–367, 2012.

[16] I. Giurgiu, O. Riva, and G. Alonso. from Clouds to Mobile Devices. pages 394–414, 2012.

[17] M. S. Gordon, D. Anoushe, J. Scott, M. Z. Morley, and M. X. Chen. Comet: Code offload by migrating execution transparently.

[18] L. Guan, X. Ke, M. Song, and J. Song. A survey of research on mobile cloud computing. In *Computer and Information Science (ICIS), 2011 IEEE/ACIS 10th Int. Conf on*, pages 387–392, May 2011.

[19] M. Hassan and S. Chen. Mobile mapreduce: Minimizing response time of computing intensive mobile applications. In J. Zhang, J. Wilkiewicz, and A. Nahapetian, editors, *Mobile Computing, Applications, and Services*, volume 95 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 41–59. Springer Berlin Heidelberg, 2012.

[20] M. A. Hassan and S. Chen. An investigation of different computing sources for mobile application outsourcing on the road. In *Mobile Wireless Middleware, Operating Systems, and Applications*, pages 153–166. Springer, 2012.

[21] S. Hill. Android 4.4 vs. ios 7 vs. windows phone 8: Which smartphone os is best? http://www.digitaltrends.com/mobile/best-smartphone-os/, 2014. Accessed: 10/13/2014.

[22] C.-C. Huang, J.-L. Huang, C.-L. Tsai, G.-Z. Wu, C.-M. Chen, and W.-C. Lee. Energy-efficient and cost-effective web API invocations with transfer size reduction for mobile mashup applications. *Wireless Networks*, 20(3):361–378, June 2013.

[23] P. Jain, R. Kabra, S. Rustagi, T. Bansal, D. Patel, and V. Raychoudhury. Mc2: On-the-fly mobile compute cloud for computational intensive task.

In *Proc. of the 5th IBM Collaborative Academia Research Exchange*, I-CARE '13, pages 7:1–7:4, New York, NY, USA, 2013. ACM.

[24] I. N. S. M. R. N. L. Jhingut, M.Z.; Ghoorun. Design and development of 3d mobile games, advances in computer-human interactions.

[25] D. Kakadia. MECCA : Mobile , Efficient Cloud Computing Workload Adoption Framework using Scheduler Customization and Workload Migration Decisions. pages 41–45, 2013.

[26] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo : a Computation Offloading Framework for Smartphones. 3.

[27] A. Khalaj and H. Lutfiyya. Handoff Between Proxies in the Proxy-Based Mobile Computing System. *2013 International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, pages 10–18, Nov. 2013.

[28] A. Khan, M. Othman, S. Madani, and S. Khan. A survey of mobile cloud computing application models. *Communications Surveys Tutorials, IEEE*, 16(1):393–413, First 2014.

[29] A. N. Khan, M. L. Mat Kiah, S. a. Madani, A. U. R. Khan, and M. Ali. Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. *The Journal of Supercomputing*, 66(3):1687–1706, June 2013.

[30] B. A. Kitchenham, D. Budgen, and O. P. Brereton. Using mapping studies as the basis for further research a participant-observer case study. *Information and Software Technology*, 53(6):638 – 651, 2011.

[31] S. Kosta, A. Aucinas, and R. Mortier. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *2012 Proc. IEEE INFOCOM*, pages 945–953. IEEE, Mar. 2012.

[32] D. Kovachev, T. Yu, and R. Klamma. Adaptive Computation Offloading from Mobile Devices into the Cloud. *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, pages 784–791, July 2012.

[33] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava. A survey of computation offl.g for mobile systems. *Mob. Netw. Appl.*, 18(1):129–140, Feb. 2013.

[34] Y.-W. Kwon and E. Tilevich. Energy-Efficient and Fault-Tolerant Distributed Mobile Execution. *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 586–595, June 2012.

[35] Y.-W. Kwon and E. Tilevich. Reducing the Energy Consumption of Mobile Applications Behind the Scenes. *2013 IEEE International Conference on Software Maintenance*, pages 170–179, Sept. 2013.

[36] K.-H. Lim and B.-D. Lee. History-based dynamic estimation of energy consumption for mobile applications. *16th International Conference on Advanced Communication Technology*, pages 714–718, Feb. 2014.

[37] Q. Liu, X. Jian, J. Hu, H. Zhao, and S. Zhang. An optimized solution for mobile environment using mobile cloud computing. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1–5, Sept 2009.

[38] M. Montesi and P. Lago. Software engineering article types: An analysis of the literature. *Journal of Systems and Software*, 81(10):1694 – 1714, 2008. Selected papers from the 30th Annual International Computer Software and Applications Conference (COMPSAC), Chicago, September 721, 2006.

[39] R. Muraleedharan. Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In *Proc. of the 2012 IEEE Symposium on Computers and Communications (ISCC)*, ISCC '12, pages 59–66, Washington, DC, USA, 2012. IEEE Computer Society.

[40] S. Murugesan and B. Venkatakrishnan. Addressing the challenges of web applications on mobile handheld devices. In *Mobile Business, 2005. ICMB 2005. International Conference on*, pages 199–205, July 2005.

[41] V. Namboodiri and T. Ghose. To cloud or not to cloud: A mobile device perspective on energy consumption of applications. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–9, June 2012.

[42] D. Niyato. A Dynamic Offloading Algorithm for Mobile Computing. *IEEE Transactions on Wireless Communications*, 11(6):1991–1995, June 2012.

[43] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *Proc. of the 12th Conf. on Evaluation and Assessment in Software Engineering*, EASE'08, pages 68–77, Swinton, UK, UK, 2008. British Computer Society.

[44] L. Pu, J. Xu, X. Jin, and J. Zhang. {SmartVirtCloud:} Virtual cloud assisted application offloading execution at mobile devices' discretion. In *2013 {IEEE} Wireless Communications and Networking Conference ({WCNC)}*, pages 4398–4403, 2013.

[45] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling interactive perception applications on mobile devices. In *Proc. of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 43–56, New York, NY, USA, 2011. ACM.

[46] A. Ravi and S. K. Peddoju. Mobility managed energy efficient Android mobile devices using cloudlet. *Proc. of the 2014 IEEE Students' Technology Symposium*, pages 402–407, Feb. 2014.

[47] J. M. Rodriguez, C. Mateos, and A. Zunino. *Energy-efficient job stealing for CPU-intensive processing in mobile devices*. Ieee, Dec. 2012.

[48] S. A. Saab, A. Chehab, and A. Kayssi. Energy efficiency in Mobile Cloud Computing: Total offloading selectively works. Does selective offloading totally work? *2013 4th Annual International Conference on Energy Aware Computing Systems and Applications (ICEAC)*, pages 164–168, Dec. 2013.

[49] A. Saarinen, M. Siekkinen, Y. Xiao, J. K. Nurminen, M. Kemppainen, and D. T. Labs. Can Offloading Save Energy for Popular Apps ?

[50] O. Shoukry and M. Fayek. Evolutionary scheduling for mobile content pre-fetching. In A.-H. Dediu, C. Martin-Vide, B. Truthe, and M. Vega-Rodriguez, editors, *Theory and Practice of Natural Computing*, volume 8273 of *Lecture Notes in Computer Science*, pages 228–239. Springer Berlin Heidelberg, 2013.

[51] Chinese/hong kong border automated with biometrics. *Biometric Technology Today*, 15(5):3 –, 2007.

[52] N. Vallina-Rodriguez and J. Crowcroft. Energy management techniques in modern mobile handsets. *Communications Surveys Tutorials, IEEE*, 15(1):179–198, First 2013.

[53] T. Verbelen, R. Hens, T. Stevens, F. De Turck, and B. Dhoedt. Adaptive online deployment for resource constrained mobile smart clients. In Y. Cai, T. Magedanz, M. Li, J. Xia, and C. Giannelli, editors, *Mobile Wireless Middleware, Operating Systems, and Applications*, volume 48, pages 115–128. Springer Berlin Heidelberg, 2010.

[54] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt. Adaptive application configuration and distribution in mobile cloudlet middleware.

[55] T. Verbelen, T. Stevens, P. Simoens, F. De Turck, and B. Dhoedt. Dynamic deployment and quality adaptation for mobile augmented reality applications. *J. Syst. Softw.*, 84(11):1871–1882, Nov. 2011.

[56] X. Wang. AppMobiCloud : Improving Mobile Web Applications by Mobile- Cloud Convergence. 2013.

[57] Y.-c. Wang, B. Donyanavard, and K.-t. T. Cheng. Energy-Aware Real-time Face Recognition System on Mobile CPU-GPU Platform. (1):1–12.

[58] F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang, and J. Ma. Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. 16(1):95–111, Oct. 2013.

[59] F. Xia, C.-H. Hsu, X. Liu, H. Liu, F. Ding, and W. Zhang. The power of smartphones. *Multimedia Systems*, Sept. 2013.

[60] S. Yang, Y. Kwon, Y. Cho, H. Yi, D. Kwon, J. Youn, and Y. Paek. Fast dynamic execution offloading for efficient mobile cloud computing. In *2013 {IEEE} International Conference on Pervasive Computing and Communications ({PerCom)}*, pages 20–28, 2013.

[61] W. Zhang, L. Chen, X. Liu, Q. Lu, P. Zhang, and S. Yang. An OSGi-based flexible and adaptive pervasive cloud infrastructure. *Science China Information Sciences*, 57(3):1–11, June 2014.

[62] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs. Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing. *Mobile Networks and Applications*, 16(3):270–284, Apr. 2011.

[63] Y. Zhang, H. Liu, L. Jiao, and X. Fu. To offload or not to offload: An efficient code partition algorithm for mobile cloud computing. In *2012 {IEEE} 1st International Conference on Cloud Networking ({CLOUDNET)}*, pages 80–86, 2012.

[64] Y. Zhou and X. Jiang. Dissecting android malware: Characterization and evolution. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 95–109, May 2012.

[65] H. Zhu, C. Huang, and J. Yan. Vulnerability evaluation for securely offloading mobile apps in the cloud. *2013 IEEE 2nd Int. Conf. on Cloud Networking (CloudNet)*, pages 108–116, Nov. 2013.