

Client Aware Scalable Cloudlet to Augment Edge Computing with Mobile Cloud Migration Service

<https://doi.org/10.3991/ijim.v14i12.14407>

Uma Nandhini D
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of
Science and Technology, Chennai, India

Udhayakumar Shanmugam (✉)
Saveetha Institute of Medical and Technical Sciences,
Chennai, Tamil Nadu, India.
mailto:udhay@gmail.com

Latha Tamiselvan
B.S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, India.

Silviya Nancy J
PES University, Bangalore, India

Abstract—Computing with mobile is still in its infancy due to its limitations of computational power, battery lifetime and storage capacity. These limitations hinder the growth of mobile computing, which in-turn affects the growth of computationally intensive applications developed for the mobile devices. So in-order to help execute complex applications within the mobile device, mobile cloud computing (MCC) emerged as a feasible solution. The job of offloading the task to the cloud data center for storage and execution from the mobile seems to gain popularity, however, issues related to network bandwidth, loss of mobile data connectivity, and connection setup does not augment well to extend the benefits offered by MCC. Cloudlet servers filled this gab by assisting the mobile cloud environment as an edge device, offering compute power to the connected devices with high speed wireless LAN connectivity. Implementation constraints of cloudlet faces severe challenges in-terms of its storage, network sharing, and VM provisioning. Moreover, the number of connected devices of the cloudlet and its load conditions vary drastically leading to unexpected bottleneck, in which case the availability to server becomes an issue. Therefore, a scalable cloudlet, Client Aware Scalable Cloudlet (CASC) is proposed with linear regression analysis, predicting the knowledge of expected load conditions for provisioning new virtual machines and to perform resource migration.

Keywords—Client aware resource scalability cloudlet, mobile cloud computing, resource migration, scalable cloudlet.

1 Introduction

Computing paradigms connected to the mobile phones have evolved from standalone mobile devices, to interactive video, then towards smart phones meant for games and social collaboration. As cloud computing evolved to offer every kind of services to the end users with minimal cost, the mobile devices connected to cloud are set to become the only device which would offer every possible kind of applications to the end users. To replace laptops and desktops and achieve that goal is not an easy task, as many issues are to be addressed. One of the major challenges ahead is with respect to network latency, real-time end to end connectivity, inter-cloud migration and traffic. On the other hand, the computational capability of the mobile devices to execute like a laptop or desktop is far from reach because of its inherent drawbacks of limited power, speed and storage. Hence to fill this gap between cloud's network delay and mobile's device inefficiency, a new sub domain has evolved namely Mobile Cloud Computing, Edge Computing and Cloudlet Computing.

1.1 Mobile cloud computing

Mobile cloud computing offers computationally intensive and storage heavy jobs to be transferred to the resource rich cloud data centers [1]. A simple architecture of the process flow is depicted in Figure 1. First, a compute intensive application request for processing to a cloud service that matches the requirements of the user's mobile application. Secondly, the task may either be offloaded completely or partially to the cloud for computation. Once it reaches the data center, they are processed and sent back to the mobile devices. This simple link between mobile and cloud have been the focus for mobile app developers mainly to store user's data in the cloud thereby relieving the utilization of mobile device. This cloud integration mechanism resolves the problem of limited processing power capabilities of mobile phones and augments it to perform better than its ability.

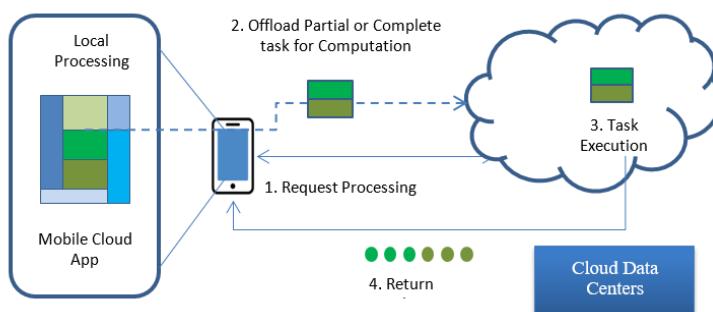


Fig. 1. Mobile cloud computing

However, the backbone of MCC relies heavily on internet to deliver the services, this poses a major issue for large size data processing, like video and image analysis.

Moreover, the cloud is a revenue model where the cost is directly proportional to the amount of data and service used. Hence any value-added services will be charges based on the utilization of cloud resources namely, the underlying virtual infrastructure, or the storage or even the application for which is the service is being offered. Therefore, to overcome the drawbacks of MCC, various new offerings have emerged, among them the most promising are the Cloudlets and Edge computing.

1.2 Edge computing

Edge computing offers its service for mobile devices which has the capability to send data to the servers that are generated by sensors from industrial machines and controllers. Data-stream acceleration for real-time data processing is achieved with no latency, thereby eliminating lag time upon which the sensors can respond immediately. Performing analytics and knowledge generation at the edge, the communication bandwidth and central control from the cloud data center is reduced. While cloud computing is a major enabler of industrial transformation, edge computing is rapidly becoming a key part of the Industrial Internet of Things (IIoT). Hence cloud and edge computing are the complementary technologies that are powering IoT. The next most important computing paradigm, that is in favor of supporting the mobile device is the cloudlet, which offers an alternative to cloud offloading using an intermediate server which is connected through the high-speed network. The server acts as a surrogate system for the cloud and connects to the distant cloud server using a wired or wireless network as presented in Figure 2.

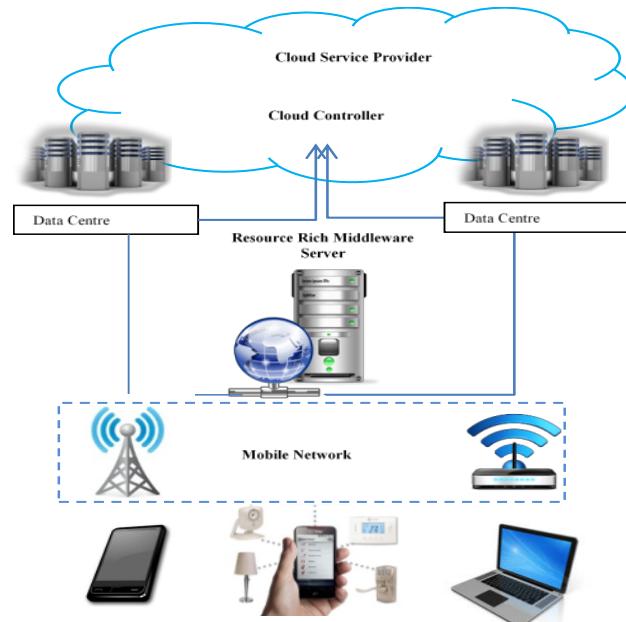


Fig. 2. Cloudlet – Cloud Connection

The wireless network is generally through a one hop Wi-Fi communication and the intermediate server is any general-purpose computing system where a micro private cloud can be setup. The concept of cloudlet can be envisioned as a local server working through Wi-Fi connectivity capable of running the workloads required by a user. Some of the major advantages of using cloudlet-based offloading over cloud are listed in Table 1. To substantiate it consider a scenario where a student in a university wishes to view a 1-hour video lecture in a particular format. Unfortunately, the video is incapable to run it in his device. Then a cloudlet inside the campus may offer to start the video by offloading it, converting it into the desired format and sending it back. Thereby, saving transfer time, processing time and battery power. To consider this cloudlet offloading it is essential to compare it with other available choices like, what if it is done within the mobile device and at the cloud server. Thus, Cloudlet is the most obvious choice to support the inefficiency of mobile devices to access cloud services. However, even at its best, the cloudlet is a small server connected to the cloud data center, hence it may have very little resource in comparison to the data center. Moreover, there are issues concerning scalability when a greater number of consumers connects to it. Therefore, the objective of the proposed work is to provide a seamless availability of cloud services within cloudlet at higher bandwidth and at less delay based on prior knowledge of the client attributes.

Table 1. Cloud Vs cloudlet offloading

Features	Cloud Offloading	Cloudlet Offloading
Server	Distant cloud data center – generally public cloud	Nearby resource rich server – private cloud
Connectivity	Either 3G or 4G Mobile internet	High speed WiFi connection based on mobile
Bandwidth	Dependent on public network with multi-path routing - low bandwidth	Private network with one-hop communication - high bandwidth
Latency	High latency due to inherent internet traffic	Very low latency because of WiFi network
Cost	High data transfer cost and cloud service cost	Nearly zero cost for data transfer and service

2 Literature Review

Improving the capabilities of mobile devices with the help of cloud platform has been the dominant scope of research where new algorithms and models are identified for offloading tasks from the mobile. Offloading can be done with two different infrastructure, namely direct cloud offloading and surrogate cloud offloading [2]. Direct offloading uses some of the major public cloud resources like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform for complex task execution. Many researchers have explored various possibilities of mobile cloud applications with the help of these public cloud infrastructures in the field of mobile healthcare, crowdsourcing and gaming [3], [4]. Though direct offloading is the best case for mobile cloud, the latency, network delay, and intermittent connection to the mobile internet

make this form of offloading a distant dream for real-time applications. Also, complex applications like video analytics and image processing may significantly affect the quality of service for the mobile consumer's end.

Indirect cloud offloading suggests using an intermediate resource to overcome the inherent problem of network failures and delays in mobile internet. Hence a pioneering vision of Mahadev Satyanarayana [5], paved the way for a surrogate system – Cloudlet, connected to the cloud to perform the task execution for mobile clients. This architecture is known as ‘cyber foraging’, acted as a bridge between the mobile device and the cloud data centers [6]. Hence a middleware-based approach emerged as an alternative for direct cloud. Based on this, many types of research have been contributed to the development of an intermediate server, as an alternate for the cloud. In the work [7], a cloudlet server offers to offload in a resource constrained edge computing environment.

Cloud scalability assures resource availability when the infrastructure that provides it becomes insufficient when handling dynamic demands [8]. Since resource scalability is one of the underlying features of cloud, new resources must be launched by adding more virtual machines [9]. Therefore, to scale the computational resources, either the existing infrastructure must be added with new hardware or the resource can be migrated it to the public cloud data center. Thus, the methodology adopted for MCC scalability must include three components, namely, the cloud hardware scalability, the network resource scalability and the scalability of mobile devices [10]. Virtual machine-based cloudlet is the best approach when unknown mobile clients interact because a VM can encapsulate the request with the transient guest environment from the permanent host of the cloudlet infrastructure [11]. Some of the challenges for a cloudlet in leveraging its server for mobile clients were described by Aleksandar [12], in that, an efficient way to enhance the productivity of cloudlet is by using persistent cache based virtual machines. These approaches though have identified that VM based cloudlet is by far the best, they have not concentrated on improving the availability when multiple requests are flooded from clients.

A recent contribution from William [13], proposes to bridge the upcoming technology of mobile edge computing with the distributed cloud by having multiple tiers of resources, serviced for clients. Also, in this research, resource management algorithm for application placement based on latency, mobility, and data center utilization have been elaborated. In another work a virtualization-based security technique on MCC is reviewed [14]. In a recent work related to scalability [15], a resource management protocol using auto-scaling methods is proposed for MCC and the futuristic Cloud Radio Access Network (C-RAN). The above literature suggest having a vertical scaling for time demanded applications and for delay tolerant applications horizontal scaling may be the choice. Thus, a reasonable scaling algorithm must be defined with scope for prediction of demand in an unknown heterogeneous environment.

3 Client Aware Scalable Cloudlet

The proposed Client Aware Resource Scalable Cloudlet (CARSC) identifies client aware features as the primary enabler to perform scalability of the cloudlet. Every client

connected to the server must send their features to the cloudlet for analysis, based on the client features the expected resource utilization can be identified through linear regression model. Based on the capabilities of the cloudlet, further resources can be provisioned within the edge devices or from remote cloud centers. Computation offloading frameworks through the surrogate system supports the novelty of mobile cloud by augmenting the capabilities of smart devices [16]. Client feature like context, location, application, time, energy, CPU utilization, security attributes are some of the prominent features that should be taken into consideration for designing a scalable cloudlet. Since the proposed scalable cloudlet is a middleware, the features of client awareness must be enforced in the design for scalability and as well as for migration. Client aware features are classified into three types, whose attributes are given in table 2.

1. Device attributes
2. Security attributes
3. Application attributes

Device Attributes: The hardware and software attributes related to the mobile devices are considered here. Hence CPU and memory utilization and its corresponding battery consumption form the most important attributes of the device. Other important features are the network capability, i.e., the type of Wi-Fi connectivity the mobile device is supporting. Based on the present scenario IEEE 802.11 b/g/n and ac are the advanced networks supported in the mobile. Hence, any device that does not match the criteria would perform with poor bandwidth quality, and this directly affects the time to offload. Apart from these, the device MAC address and its hardware features like the battery type and capacity, number of cores and memory contribute to it.

Application Attributes: The type of application which are working online and its size helps to judge the performance of the device. Moreover, change in size of the application is one important attribute to identify if there is any malware being injected into the software. Apart from that, the memory utilized by the application, number of threads required to run the application, and the total number of times the applications have offloaded are also considered.

Table 2. Client aware attributes

Device Attributes	Application Attributes	Security Attributes
CPU utilization	Type of Application	Security Certificates
Memory Utilization	Size of Application	Malware attacks
Battery Power	Memory Utilized	Trusted Credentials
MAC address	No. of Threads per App.	Frequency of Access
Hardware Configurations	No. of times offloaded	Geographical Location
Network Capability	Apps used through Wi-Fi	Privacy Constraints

Security Attributes: Security is an essential part of any device and an environment like CASC, where many clients may interact with the cloudlet for services. Hence to provide a service the client attributes like malware attack history, the credential store to prove the trustworthiness of the clients and its privacy constraints are a factor that must be taken into consideration. Any certificate received from the third party would

necessarily enhance the security of a device because only for a secure device, a third party would assess and provide a connectivity to offload. Similarly, history of malware attacks will state the vulnerability of the device [17]. Trust credentials stored in the trust store would also prove the device trustworthiness. Finally, the location and access patterns criteria help to judge the client attribute with respect to security.

3.1 Scalable cloudlet model

Augmentation of the mobile is done when the application currently running in the mobile requires external support for accelerated execution. The external support may be either the cloudlet or the cloud servers. To decide when to offload must be taken based on the ability of the device to do the job given the present client context. Hence an offloading decision algorithm is developed, which uses the attributes of the client and the availability of the required application at the cloudlet as presented in Figure 4. This process of judging the ability of the system is done by the Client Aware Cloudlet (CAC) by making a preliminary connection establishment.

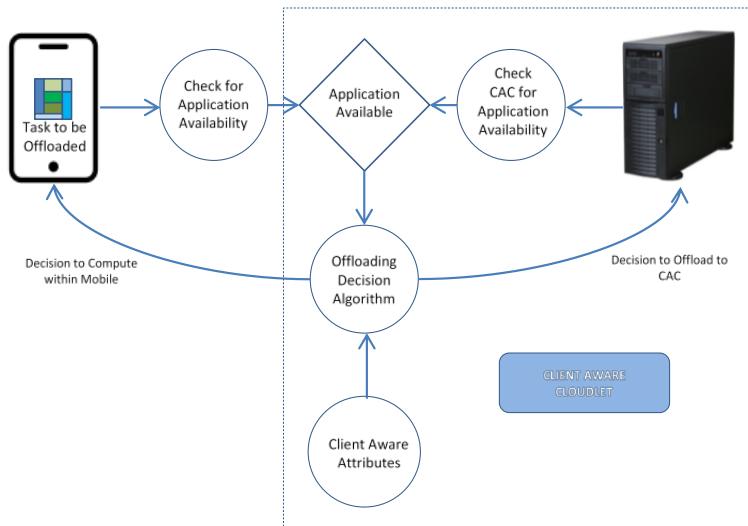


Fig. 3. Scalable cloudlet architecture

The Cloudlet is designed as a micro private cloud implemented on commodity hardware with very limited resources in comparison to a cluster of servers as depicted in figure 3. This configuration is adopted to prove that the proposed model can be setup with limited resources by anyone to contribute his resources for offloading service.

3.2 Need for resource scalability

Under changing workloads, the CAC must be stable and manage the increasing demands raised by the addition of new offloading task from existing mobile users and as

well from new users [18]. Therefore, to meet this demand it is essential to add new resources, hence the choice of resources can be from 3 different sources namely:

1. Scalability through elasticity of the existing CAC if additional VMs can be provisioned.
2. Additional resources can be provisioned by converting the micro private cloud into a mini-private cloud or into a full-fledged cluster of computers so that 100s of VM are ready to serve.
3. The final approach would be extending the CAC to connect with any available cloud server who can able to run the required application at remote data center.

The first two approaches can be implemented within the perimeter of the source and hence our work is developed to achieve scalability by extending the existing resources based on the scalability protocol.

4 Implementation of CASC

The client aware scalable cloudlet requires atleast 2 VMs are needed for running the client requesting applications and CAC components. Hence the need to start new VMs if available within or from its cluster is essential when there is an unexpected number of mobile clients' request for offloading their task. The launching of new VMs when there is a load imbalance at the CAC is governed by Client Aware Resource Scalability (CARS) protocol as presented in figure 4. Moreover, CAC being an exchange point at the edge of the internet, the private cloud needs to coordinate with the public cloud, making this into a hybrid cloud environment if the need arises [19].

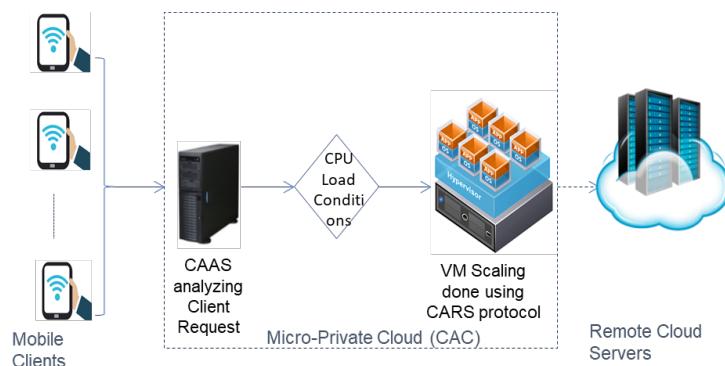


Fig. 4. Resource scaling at CAC using CARS protocol

Though hybrid VM scalability is considered as a future scope of the research, it is one of the potential areas to explore. This is because; the idea of cloudlet is fast becoming an exchange platform to the cloud that is currently evolving as an upcoming computing paradigm called edge computing.

4.1 Client aware resource scalability protocol

The client aware resource scalability protocol identifies the history of past interactions of the mobile clients and its corresponding load as the important client aware features. For example, in the video conversion process, the load for converting a video file depends on the size of the file. The history of that load is dependent on the number of access made by those mobile users. Hence if the same user is arriving at time $a(t+1)$, then it can be assumed that a similar load can be expected given the past value of the same user. Therefore, the load which is nothing but the CPU utilization of the running virtual machine is to be calculated. Thus, based on the evidence and likelihood a look-ahead approach for predicting the load is carried out using the auto scaling functionalities inbuilt in the OpenStack's Heat component. This functionality is fine-tuned to predict the expected workload and initiate the VMs when the load exceeds the threshold. The upper and lower thresholds for virtual machine scaling are estimated based on multiple trials for the different input condition. The CPU utilization of the CAC when multiple mobile client's access are tested and the values are mapped for a time period of 3000 sec as depicted in the graph of Figure 5.

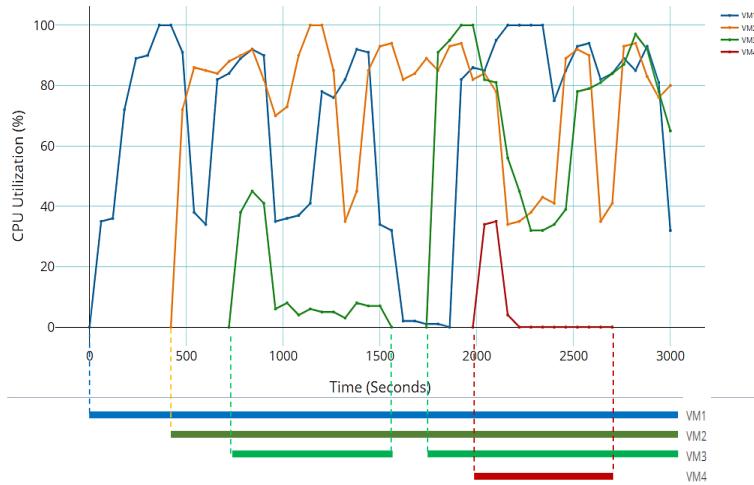


Fig. 5. CPU utilization Graph for 4 different VMs within CASC

It can be seen from the graph and its related table 3, that at time 0 the CPU utilization of VM1 is also 0 and as the 1st client uses the server for computation, it uses around 30 seconds for the first file of 10MB size. As the 2nd client with file size 15 MB and 3rd with 21MB uses VM1 concurrently it reaches beyond 80%. The upper threshold for CPU utilization is fixed at around 80% for more than 3 minutes. During this time few more files namely 4th, 5th and 6th files are converted thereby reaching the peak of 100%. Hence after 3 minutes, the 2nd VM gets initialized for resource scalability, which starts at around 480 seconds. Similarly, the VM3 starts at 780 seconds, and has its lower bound at 25% or 10 minutes, then its shuts down the VM3 at 1500th second.

Table 3. Statistics of VM's in CASC

Time in Seconds	CPU Utilization in %				Total No. of VM Currently active.
	VM 1	VM 2	VM3	VM4	
0	0				
60	35				1
120	36				1
180	72				1
240	89				1
300	90				1
360	100				1
420	100	0			1
480	91	72			2
540	38	86			2
600	34	85			2
660	82	84			2
720	84	88	0		2
780	89	90	38		3
840	92	92	45		3
900	90	82	41		3
960	35	70	6		3
1020	36	73	8		3
1080	37	90	4		3
1140	41	100	6		3
1200	78	100	5		3
1260	76	85	5		3
1320	82	35	3		3
1380	92	45	8		3
1440	91	85	7		3
1500	34	93	7		3
1560	32	94	0		2
1620	2	82			2
1680	2	84			2
1740	1	89	0		2
1800	1	85	91		3
1860	0	93	95		3
1920	82	94	100		3
1980	86	82	100	0	3
2040	85	84	82	34	4
2100	95	78	81	35	4
2160	100	34	56	4	4
2220	100	35	45	0	4
2280	100	38	32	0	4
2340	100	43	32	0	4
2400	75	41	34	0	4
2460	85	89	39	0	4
2520	93	92	78	0	4
2580	94	90	79	0	4
2640	82	35	81	0	4
2700	84	41	84	0	4
2760	89	93	87		3
2820	85	94	97		3
2880	93	83	92		3
2940	81	76	78		3
3000	32	80	65		3

Observation of CARS: Based on the CA resource scalability protocol the CAC efficiently provided the necessary service without any interruption. The following are the estimated CPU load conditions when offloading begins.

Mobile Device 1 - CPU utilization around 30 to 40 %

Mobile Device 2 - CPU utilization around 75 – 90 %

Mobile Device 3 - CPU utilization > 90%

Hence for heavy duty applications like video conversion, only 2 to 3 consecutive conversion operations could be performed in a single VM, hence when the request is greater than 3 then additional VMs are instantiated based on the threshold conditions of CARS. Thus, VM scalability is an essential process while designing the framework for CAC.

Algorithm 1: CA Resource Scalability Protocol

```
while application streaming at CAC
{
    get (No. of mobile Users, No. of Application Instance
running, No. of Application Service in Queue)
    for VM(p)
        Check CPU utilization in %;
        Define Upper Threshold = 80 %;
        Lower Threshold = 25%;
    {
        if CPU > 80% for more than 3 min
            then upscale and Initiate VM(k)
        else if CPU <25% for > than 10min
            then descale VM(p)
    }
}
```

5 Cloud Migration Service

The concept of CAC is to utilize the available resources within the micro-private cloud environment for minimizing the cost and response time. This could well be achieved if the number of requests is minimal at any given time and if the requested service within the scope of availability. However, with the resource of a dual-core processor with 8GB memory, the CAC server will be of no use when there is a surge in request for processing High Definition (HD) videos or performing 3D rendering. Hence there must be always a cloud supported resources for execution that can able to support resource migration to external virtual machines.

A Client Aware Migration Service (CAMS) is deployed within the OpenStack middleware, for resource migration to the remote cloud datacenter. The CAMS already acts

as an exchange service for helping the recommendation engine for downloading and deploying recommended applications into the application server.

However, with respect to resource scalability, it performs migration of tasks based on the information provided by CARM via the internet to the cloud and completes the task, and stores it at the CAC. If the client's context changes then based on CAAS the output can be further sent to the cloud storage. Thus, CAMS is an essential service as long as the CAC is self-sufficient in terms of applications and resources.

6 Conclusion

Offloading computation from a resource poor mobile device to a surrogate machine has been the major focus towards integrating mobile and cloud computing. Despite being a promising technology the level of adoption and full-fledged integration on a client device is rather slow, because of its inability to satisfy the client requirements. One of the major requirements is the stability of the Cloudlet, which has been effectively addressed by incorporating a scalability protocol CARS for the scalable cloudlet. This result has shown that within a simple standalone setup of micro private cloud, it is possible to scale and handle multiple clients by initiating VMs when required. Also, the VM can be activated and deactivated based on the client aware features of the mobile devices that are connected to the CASC server. Thus, the proposed work envisages a new paradigm where mobile cloud supports the clients not by sending the service to be offloaded to the distant cloud but to a nearby server at the edge of the cloud. Hence this paper introduces a method to bring in Mobile Cloud Computing and Edge Computing to serve the client mobile device based on its client aware features.

7 References

- [1] M Beck, W Hao, and A Campan. (2017). Accelerating the Mobile Cloud: Using Amazon Mobile Analytics and k-means Clustering. IEEE Computing and Communication Workshop and Conference, pp. 1-7. <https://doi.org/10.1109/ccwc.2017.7868372>
- [2] Nur Idawati Md Enzai, and Maolin Tang. (2014). A Taxonomy of Computation Offloading in Mobile Cloud Computing. IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, pp. 19-28. <https://doi.org/10.1109/mobilecloud.2014.16>
- [3] H. Wu, Q. Wang, and K. Wolter. (2013). Mobile Healthcare Systems with Multi-cloud Offloading. IEEE 14th International Conference on Mobile Data Management, pp. 188-193. <https://doi.org/10.1109/mdi.2013.92>
- [4] Poovam Raj T.T, UdhayaKumar S and Silviya Nancy J. (2017). Smart City Based Mobile Application for Seamless Communication of Differently-Abled. International Journal of Multimedia and Ubiquitous Engineering, Vol.12, No.1, pp. 177-190. <https://doi.org/10.14257/ijmue.2017.12.1.15>
- [5] Mahadev Satyanarayanan, Bahl P, Caceres R, and Nigel Davies. (2009). The Case for VM-Based Cloudlets in Mobile Computing, IEEE Pervasive Computing, Vol. 8:4, pp. 14-23. <https://doi.org/10.1109/mprv.2009.64>

- [6] Jason Flinn. (2012). Cyber Foraging: Bridging Mobile and Cloud Computing. *Synthesis Lectures on Mobile and Pervasive Computing*, Vol. 7:2, pp. 1-103. <https://doi.org/10.2200/s00447ed1v01y201209mpc010>
- [7] G. A. Lewis, S. Echevería, S. Simanta, B. Bradshaw, and J. Root. (2014). Cloudlet-based Cyber-Foraging for Mobile Systems in Resource Constrained Edge Environments, 36th International Conference on Software Engineering, pp. 412-415. <https://doi.org/10.1145/2591062.2591119>
- [8] Yating Wang, Ing-Ray Chen, and Ding-Chau Wang. (2015). A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges. *International Journal of Wireless Personal Communications*, Vol. 80:4, pp. 1607-1623. <https://doi.org/10.1007/s11277-014-2102-7>
- [9] Huber Flores and S. N. Srivama. (2014). Mobile Cloud Middleware. *Journal of Systems and Software*, Vol. 92, pp. 82-94. <https://doi.org/10.1016/j.jss.2013.09.012>
- [10] Zhiyuan Yin, F. R. Yu, and Shengrong Bu. (2016). “Mobile Cloud Computing with Telecom Operator Cloud. *Advances in Mobile Cloud Computing Systems*, Taylor and Francis Publications, pp. 1-40. <https://doi.org/10.1201/b19128-2>
- [11] Uma Nandhini and Latha Tamilselvan. (2014). Computational Analytics of Client Awareness for Mobile Application Offloading with Cloud Migration. *KSII Transactions on Internet and Information System*, Vol. 8:11, pp.3916-3936. <https://doi.org/10.3837/tiis.2014.11.014>
- [12] G. Lewis, M. Satyanarayanan, S. Simanta, and K. Ha. (2011). Cloud Offload in Hostile Environments. Carnegie Mellon University, NSF Grant No. CNS-0833882
- [13] Aleksandar B and Marjan Gusev. (2014). Cloudlet Challenges. *International Symposium on Intelligent Manufacturing and Automation, Procedia Engineering*, Vol. 69 pp. 704-711. <https://doi.org/10.1016/j.proeng.2014.03.045>
- [14] Boubakeur Annane, Osman Ghazali. (2019). Virtualization-Based Security Techniques on Mobile Cloud Computing: Research Gaps and Challenges. *International Journal of Interactive Mobile Technologies*. Vol. 13:4, pp. 20-32. <https://doi.org/10.3991/ijim.v13i04.10515>
- [15] William Tarneberg, Amardeep Mehta, Eddie Wadbro, Johan Tordsson, Johan Eker, Maria Kihl, and Erik Elmroth. (2017). Dynamic Application Placement in the Mobile Cloud Network. *Future Generation Computer Systems*, Vol. 70, pp. 163-177. <https://doi.org/10.1016/j.future.2016.06.021>
- [16] Mais Haj Qasem, Alaa Abu-Srhan, Hutaif Natoureh, Esra Alzaghouli. (2020). Fog Computing Framework for Smart City Design. *International Journal of Interactive Mobile Technologies*. Vol. 14:1 109-125. <https://doi.org/10.3991/ijim.v14i01.9762>
- [17] Tanweer Alam, Mohamed Benaïda. (2018). CICS: Cloud–Internet Communication Security Framework for the Internet of Smart Devices. *International Journal of Interactive Mobile Technologies*. Vol. 12:6. 74-84. <https://doi.org/10.3991/ijim.v12i6.6776>
- [18] C. S. Magurawalage, Kun Yang, Ritosa Patrik, Michael Georgiades, and Kezhi Wang. (2017). A Resource Management Protocol for Mobile Cloud Using Auto-Scaling. *Distributed, Parallel, and Cluster Computing*, Cornell University Library.
- [19] Uma Nandhini, Latha Tamilselvan, Silviya Nancy J, Udhaya Kumar S. (2017). A Secure Client Aware Certification for Mobile Cloud Offloading Decision. *International Journal of Intelligent Engineering and Systems*, Vol. 10:2

8 Authors

Uma Nandhini D is a faculty member in the Department of Computer Science and Engineering at Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, a Deemed to be University in Chennai, Tamil Nadu, India. She received her Ph.D in the domain of Mobile Cloud Computing at B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai. She has around 8 International Journals which are scopus indexed and 1 Web of Science Indexed Journals to her credit. Her area of interest includes, mobile cloud, machine learning, and data mining. E.mail: umaudhay@gmail.com, drumanandhini@veltech.edu.in

Udhayakumar Shanmugam presently working as Professor, in the Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India. He received his Ph.D in the domain of Trusted Cloud Computing at B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai. He has presented more than 55 International conferences and has to his credit 15 International Scopus indexed journals. His area of research includes Cloud Security, Augmented Reality, Deep Learning, and Edge Analytics. E.mail: mailtoudhay@gmail.com, udhayakumars.sse@saveetha.com

Latha Tamiselvan is a Professor in the Department of Information Technology and Director (Data Center), at B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai. She completed her Ph.D in the domain of Wireless Networks at Anna University, Chennai, India. She has around 17 International Journals and presented 27 International Conferences to her credit. She has guided and completed 5 Ph.D Students and her area of research includes, Cloud Computing, Wireless Networks and Network Security. E.mail: latha.tamil@crescent.education

Silviya Nancy J is a faculty member in the Department of Computer Science and Engineering, PES University, Bangalore, India. She has completed her Master's Degree at Anna University. She has around 10 International Journals and presented 5 International conferences. Her area of research includes Artificial Intelligence, Cloud Computing, and Data Science. E.mail: jsilviyanancy@gmail.com

Article submitted 2020-03-25. Resubmitted 2020-04-16. Final acceptance 2020-04-20. Final version published as submitted by the authors.