Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges

Saeid Abolfazli, *Member, IEEE,* Zohreh Sanaei, *Member, IEEE,* Ejaz Ahmed, *Member, IEEE,* Abdullah Gani, *Senior Member, IEEE*, Rajkumar Buyya, *Senior Member, IEEE*

***Abstract*—Recently, Cloud-based Mobile Augmentation (CMA) approaches have gained remarkable ground from academia and industry. CMA is the state-of-the-art mobile augmentation model that employs resource-rich clouds to increase, enhance, and optimize computing capabilities of mobile devices aiming at execution of resource-intensive mobile applications. Augmented mobile devices envision to perform extensive computations and to store big data beyond their intrinsic capabilities with least footprint and vulnerability. Researchers utilize varied cloud-based computing resources (e.g., distant clouds and nearby mobile nodes) to meet various computing requirements of mobile users. However, employing cloud-based computing resources is not a straightforward panacea. Comprehending critical factors (e.g., current state of mobile client and remote resources) that impact on augmentation process and optimum selection of cloud-based resource types are some challenges that hinder CMA adaptability. This paper comprehensively surveys the mobile augmentation domain and presents taxonomy of CMA approaches. The objectives of this study is to highlight the effects of remote resources on the quality and reliability of augmentation processes and discuss the challenges and opportunities of employing varied cloud-based resources in augmenting mobile devices. We present augmentation definition, motivation, and taxonomy of augmentation types, including traditional and cloud-based. We critically analyze the state-of-the-art CMA approaches and classify them into four groups of distant fixed, proximate fixed, proximate mobile, and hybrid to present a taxonomy. Vital decision making and performance limitation factors that influence on the adoption of CMA approaches are introduced and an exemplary decision making flowchart for future CMA approaches are presented. Impacts of CMA approaches on mobile computing is discussed and open challenges are presented as the future research directions.**

***Index Terms*—Cloud-based Mobile Augmentation, Mobile Cloud Computing, Cloud Computing, Resource-intensive Mobile Application, Computation Offloading, Resource Outsourcing.**

I. Introduction

SINCE a decade ago, the visions of *'information under fingertip'* [1] and *'unrestricted mobile computing'* [2] aroused the need to enhance computing power of mobile devices to meet the insatiable computing demands of mobile users [3]. In the late 90s, the concept of load sharing and remote execution aimed to augment computing capabilities of mobile devices by shifting the resource-intensive mobile codes to surrogates (powerful computing device(s) in vicinity) [4]–[6]. Although remote execution efforts [7]–[18] have yielded many impressive achievements, several challenges such as reliability, security, and elasticity of surrogates hinder the remote execution adaptability [19]. For instance, the resource sharing and computing services of surrogates can be terminated without prior notice and their content can be accessed and altered by the surrogate machine or other users in the absence of a Service Level Agreement (SLA). SLA is a formal contract employed and negotiated in advance between service provider and consumer to enforce certain level of quality against a fee.

Few years later, emergence of cloud resources created an opportunity to mitigate the shortcomings of utilizing surrogates in augmenting mobile devices. Cloud is a type of distributed system comprised of a cluster of powerful computers accessible as unified computing resource(s) based on an SLA [20]. Cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service or service provider interaction" [21] stimulates researchers to adopt the cutting edge technology in mobile device augmentation: Cloud-based Mobile Augmentation (CMA). Cloud-based Mobile Augmentation (CMA) is the-state-of-the-art mobile augmentation model that leverages cloud computing technologies and principles to increase, enhance, and optimize computing capabilities of mobile devices by executing resource-intensive mobile application components in the resource-rich cloud-based resources. Cloud-based resources include varied types of mobile/immobile computing devices that follow cloud computing principles [22], [23] to perform computations on behalf of the resource-constraint mobile devices. Figure 1 depicts major building blocks of a typical CMA system. It is notable that these building blocks are optional superset, and specific CMA system may not have all these building blocks.

CMA efforts [24]–[27], [27]–[49] exploit various cloud-based computing resources, especially distant clouds and proximate mobile nodes to augment mobile devices. Distant clouds are giant clouds such as Amazon EC2[1] located inside the vendor premise —far away from the mobile clients—offering infinite, elastic computing resources with extreme computing

Manuscript received Dec 18, 2012; revised March 05, 2013 and 06 May, 2013;This work is funded by the Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant - UM.C/HIR/MOHE/FCSIT/03. Ejaz Ahmed's research work is supported by the Bright Spark Unit, University of Malaya, Malaysia.

Saeid Abolfazli(corresponding author), Zohreh Sanaei, Ejaz Ahmed, and Abdullah Gani are with the Department of Computer System & Technology, The University of Malaya, Kuala Lumpur, Malaysia (e-mail: {abolfazli,sanaei, ejazahmed}@ieee.org; abdullah@um.edu.my)

RajKumar Buyya is with the Department of Computing and Information Systems, The University of Melbourne, 111, Barry Street, Carlton, Melbourne, VIC 3053, Australia, Email: raj@csse.unimelb.edu.au

[1]http://aws.amazon.com/ec2/

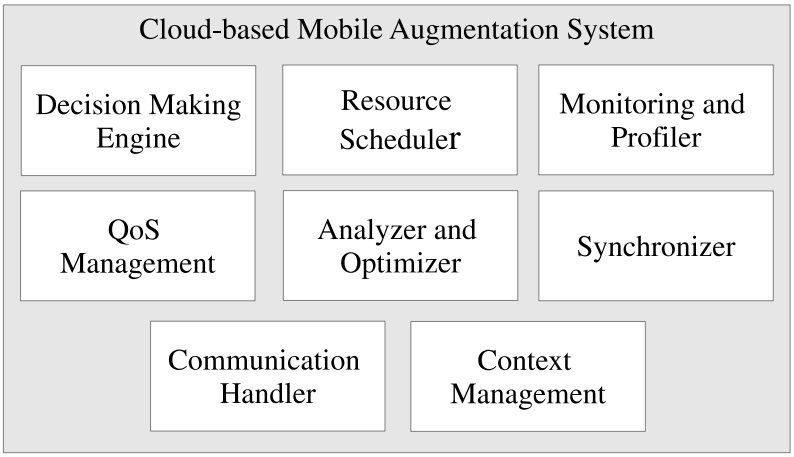

Fig. 1. Major Building Blocks of an Exemplary CMA System.

power and high WAN (Wide Area Network) latency. Proximate mobile nodes are building a cluster of mobile computing devices scattered near the mobile clients offer limited computing power with lower WAN latency than distant clouds.

Although heterogeneity among cloud-based resources increases service flexibility and enhances users' computing experience, determining the most appropriate computing resources among available options and performing upfront analysis of influential factors (e.g., user preferences and available native mobile resources) are critical in the adaptability of CMA approaches. Thus, 'resource scheduler' and 'analyzer and optimizer' components depicted in Figure 1 are needed to analyze and allocate appropriate resources to each task in a typical CMA system. Moreover, several questions need to be addressed before the CMA concept can be successfully employed in the real scenarios. For instance, can CMA augment computing capabilities of mobile devices and save local resources to enhance user experience? Is CMA always feasible and beneficial? Which type of resources is appropriate for a certain task? Answering these questions requires 'monitoring and profiler', 'QoS management', 'context management', and 'decision making engine' components to perform in each CMA system (see Figure 1). Therefore, an augmentation decision engine similar to those used in [25], [33], [49] and exemplary decision making flow presented in this paper (discussed in part VI-C) to determine the mobile augmentation feasibility is needed to amend the CMA performance and reliability. During augmentation process, the local and native application state stack needs synchronization to ensure integrity between native and remote data. Upon successful outsourcing, remote results need to be returned and integrated to the source mobile device. Thus, the 'Synchronizer' component needs to perform in typical CMA approaches (see Figure 1).

Although CMA approaches can empower mobile processing and storage capabilities, several disadvantages such as application development complexity and unauthorized access to remote data demand a systematized plenary solution. Performance of the CMA systems is highly influenced by various challenges and issues of wireless networking and cloud computing technologies. CMA researchers require a high performance, elastic, robust, reliable, and foreseeable communication throughput between mobile nodes and cloud servers which is not yet realized despite of remarkable efforts and achievements of communication and networking societies. Current shortcomings and deficiencies of wireless communication and networking, especially seamless connectivity and mobility, high performance communication throughput provisioning, and wireless data interception discourage system analysts, engineers, developers, and entrepreneurs from deploying CMA-enabled mobile applications due to the high risk of system malfunction and user experience degradation.

Moreover, CMA systems require accurate estimation mechanisms to predict the overall time and energy consumption of communication and computation tasks while exploiting clouds. Such estimation is a challenging task considering huge infrastructures' performance diversity [50] and policy heterogeneity [51] of cloud services in intermittent wireless environment. Despite of blooming efforts endeavoring to analyze and comprehend the cloud computing model and behavior [52]–[55], CMA solutions are still unable to accurately foresee required time and energy of exploiting cloud resources to execute intensive applications. Additionally, sundry cloud challenges, especially live VM migration, infrastructure and platform heterogeneity, efficient allocation of clouds to tasks, QoS management, security, privacy, and trust in cloud increase system complexity and decrease successful CMA systems adoption.

Among limited studies of the domain, [19] and [56] survey remote execution and application offloading algorithms with focus on how task offloading is performed in various efforts. Fernando et al. [57] and Dinh et al. [58] sought to explain the convergence of mobile and cloud computing, and distinguish it from the earlier domains such as cloud and grid computing [59]. The authors describe issues, particularly mobile application offloading, privacy and security, context awareness, and data management. Sanaei, Abolfazli, Gani, and Buyya [51] present a comprehensive survey on MCC with major focus on heterogeneity. The authors describe the challenges and opportunities imposed by heterogeneity and identify hardware, platform, feature, API, and network as the roots of MCC

TABLE I
LIST OF ACRONYMS AND CORRESPONDING FULL FORMS.

Acronym	Full form
2D	2 Dimensional
2G	2nd Generation
3D	3 Dimensional
3G	3rd Generation
API	Application Programming Interface
App	Application (mobile application)
ARM	Advanced RISC Machines
CMA	Cloud-based Mobile Augmentation
CPU	Central Processing Unit
DSL	Domain Specific Language
DVMS	Dynamic VM Synthesis
FTP	File Transfer Protocol
GPU	Graphics Processing Unit
GUI	Graphical User Interface
I/O	Input/Output
IaaS	Infrastructure as a Service
IP	Internet Protocol
IP TV	Internet Protocol Television
iSCSI	Internet Small Computer System Interface
MCC	Mobile Cloud Computing
MNO	Mobile Network Operator
OS	Operating System
P2P	Peer-to-Peer
PC	Personal Computer
QoS	Quality of Service
R&D	Research and Development
RAM	Random Access Memory
RISC	Reduced Instruction Set Computing
RPC	Remote Procedure Call
SAL	Service Abstraction Layer
SLA	Service Level Agreement
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery and Integration
UI	User Interface
VM	Virtual Machine
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity

heterogeneity. They explain major heterogeneity handling approaches, particularly virtualization, service oriented architecture, and semantic technology. However, the computing performance, distance, elasticity, availability, reliability, and multi-tenancy of remote resources are marginally discussed in these studies that necessitate further research to explain the impact of remote resources on augmentation process and highlight paradigm shift from the unreliable surrogates to reliable clouds.

In this paper, we survey the state-of-the-art mobile augmentation efforts that employ cloud computing infrastructures to enhance computing capabilities of resource-constraint mobile devices, especially smartphones. To the best of our knowledge, this is the first effort that studies the impacts of cloud-based computing resources on mobile augmentation process. We differentiate augmentation from similar concepts of load sharing and remote execution, and present augmentation motivation. We review efforts that endeavor to mitigate the mobile devices' shortcomings and classify them as hardware and software to devise a taxonomy. The impacts of CMA in mobile computing are presented. The characteristics of cloud-based remote resources and their role in CMA effectiveness are studied and classified into four groups, namely distant immobile clouds, proximate immobile computing entities, proximate mobile computing entities, and hybrid based on their mobility and physical location traits. Furthermore, the state-of-the-art CMA models are reviewed and taxonomized into four classes of distant fixed, proximate fixed, proximate mobile, and hybrid according to our cloud-based resource classification. Factors impact on the CMA adaptability are identified and described as augmentation environment, user preferences and requirements, mobile devices, cloud servers, and contents. Five major metrics that limit the performance of CMA approaches are studied. A sample flowchart of decision making engines for imminent CMA solutions is presented and several open challenges are discussed as the future research directions. Such survey is beneficial to the communication and networking communities, because comprehending CMA process and current deployment challenges are beneficial in modifying the fundamental networking infrastructures to optimize the CMA process. In this paper, we use the terms mobile devices and smartphones interchangeably with similar notion. Table I shows the list of acronyms used in the paper.

The remainder of this paper is organized as follows. Section II introduces mobile computation augmentation, presents its motivation and describes the taxonomy of mobile augmentation types. The impacts of CMA on mobile computing are presented in Section III. Section IV presents the analysis and taxonomy of varied cloud-based augmentation resources. Comprehensive survey of the state-of-the-art CMA approaches is presented and taxonomy is devised in Section V. We discuss the CMA decision making and limitation factors and illustrate CMA feasibility in Section VI. Finally, open research challenges are presented in Section VII and paper is concluded in Section VIII.

II. MOBILE COMPUTATION AUGMENTATION

In this Section, we present a definition on mobile computing augmentation based on the available definitions on the relevant concepts, particularly remote execution [5] and cyber foraging [6]. Additionally, the motivation for performing mobile computation augmentation is described and taxonomy of mobile augmentation types is presented.

TABLE II
INITIAL FEATURES OF MOBILE EMPOWERMENT APPROACHES.

Approach	Architecture	Client Load	Migration	Partitioning	Server	Mobility
Load Sharing	Client-Server	Entire Task	Entire task	NA	Server	NA
Remote Execution	Client-Server	Entire Task	Entire/partial	Static	Server /desktop	No
Cyber Foraging	Client-Server Peer-to-Peer	Entire Task	Entire/partial	Dynamic	Surrogates	No
Mobile Computation Augmentation	Varies, e.g., Client-server P2P, Adhoc collaborative	Entire/partial /Nil	Entire/partial/ Nil migration (Use remote services)	Static & dynamic	Server, surrogate &mobile	Yes

A. Definition

Indeed, empowering computation capabilities of mobile devices is not a new concept and there have been different approaches to achieve this goal, including load sharing [4], remote execution [5], cyber foraging [6], and computation offloading [60], [61] that are described as follows. We have analyzed them and summarized the analysis results in Table II. Results in this Table are extracted from the early efforts in each category, which are already deviated from their original characteristics due to the research achievements.

• **Load Sharing:** Othman and Hailes' work [4] in 1998 can be considered as one of the earliest efforts to conserve native resources of mobile devices using a software approach. The main idea is inspired from the concept of load balancing in distributed computing that is "a strategy which attempts to share loads in a distributed system without attempting to equalize its load" [4]. This approach migrates the whole computation job for remote execution. It considers several metrics such as job size, available bandwidth, and result size to identify if the load balancing and transferring the job to the remote computer can save energy. However, they need to send the task and data to the nearest base station and wait for the results to return. The base station is responsible to find appropriate server to run the job and forward the results back to the mobile device. Moreover, computing server is a fixed computer and there is no provision for user and code mobility at run time.

• **Remote Execution:** The concept of remote execution for mobile clients emerged in 90s and several researchers [5], [62]–[65] endeavor to enable mobile computers to performing remote computation and data storage to conserve their scarce native resources and battery. In 1998 [5], feasibility of remote execution concept on mobile computers, particularly laptops was investigated. The authors report that remote execution can save energy if the remote processing cost is lower than local execution. Remote execution involves migrating computing tasks from the mobile device to the server prior the execution. The server performs the task and sends back the results to the mobile device. However, difference between computation power of client and server is not a metric of decision making in this method. Moreover, the whole task needs to be migrated to the remote server prior the execution which is an expensive effort. It also neglects the impact of environment characteristics on the remote execution outcome. Static decision making is another shortcoming of this proposal.

• **Cyber Foraging:** Satyanarayana in 2001 [6] further developed the remote execution idea by considering dynamism in remote execution process. The author defined cyber foraging as the process "to dynamically augment the computing resources of a wireless mobile computer by exploiting wired hardware infrastructure". Resources in cyber foraging are stationary computers or servers in public places —connected to wired Internet and power cable—that are willing to perform intensive computation on behalf of the resource-constraint mobile devices in vicinity.

However, load sharing, remote execution, and cyber foraging approaches assume that the whole computing task is stored in the device and hence, it requires the intensive code and data to be identified and partitioned for offloading —either statically prior the execution or dynamically at runtime —which impose large overhead on resource-poor mobile device [19]. Moreover, as Kumar et al. [66] explain, for each mobile user that runs the intensive application, the whole offloading process must be repeated including decision making process in the device and transferring the heavy components and large data to the network. Due to slight differences among these concepts, researchers use the terms 'remote execution', 'cyber foraging', and 'computation offloading' interchangeably in the literature with similar principle and notion.

Nevertheless, researchers in recent activities [36], [42], [45], [46] aim to enhance computing capabilities of mobile devices in a slightly different manner. They assume to store the intensive code and data outside the device and keep the rest in the mobile device instead of storing the whole task —including both lightweight and intensive code and data —in the mobile device. Therefore, the overhead of identifying, partitioning, and migrating the resource-intensive task is mitigated, energy is saved, and storage problem is alleviated in mobile devices. Moreover, storing intensive components outside the device, in a publicly accessible storage, can increase their reusability and enable more than one user to leverage their computation services. Therefore, we coin the term *mobile computation augmentation* as the wider phrase that subsumes load sharing, remote execution, cyber foraging, and other approaches that

augment computing capabilities of mobile devices.

• **Mobile Computation Augmentation:** Mobile computation augmentation, or augmentation in brief, is the process of increasing, enhancing, and optimizing computing capabilities of mobile devices by leveraging varied feasible approaches, hardware and software. Mobile device is any non-stationary, battery-operating computing entity able to interact with end-user and execute transactions, store data, and communicate with the environment using wireless technologies and varied sensors. Smartphone, Tablet, handheld/wearable computing devices, and vehicle mount computers are mobile device instances. Approaches that can augment mobile devices include hardware and software. Hardware approach involves manufacturing high-end physical components, particularly CPU, memory, storage, and battery. Software approaches can be —but are not limited to —computation offloading, remote data storage, wireless communication, resource-aware computing, fidelity adaptation, and remote service request (e.g., context acquisition).

Augmentation approaches can increase computing capabilities of mobile devices and conserve energy. They can be leveraged in three main categories of applications, namely (i) computing-intensive software such as speech recognition and natural language processing, (ii) data-intensive programs such as enterprise applications, and (iii) communication-intensive applications such as online video streaming applications. The augmented mobile device is able to perform complex tasks that could not otherwise perform. Hence, the mobile application developers do not take into account resource shortcomings of mobile devices in developing mobile application and users will not consider their devices weaknesses in utilizing varied complex applications.

B. Motivation

Mobile devices have recently gained momentous ground in several communities like governmental agencies, enterprises, social service providers (e.g., insurance, Police, fire departments), healthcare, education, and engineering organizations [67], [68]. However, despite of significant improvement in mobiles' computing capabilities, still computing requirements of mobile users, especially enterprise users, is not achieved.

Several intrinsic deficiencies of mobile devices encumber feasibility of intense mobile computing and motivate augmentation. Leveraging augmentation approaches, vision of performing intense mobile operations and control such as remote surgery, on-site engineering, and visionary scenarios similar to the lost child and disaster relief described in [69] will become reality. In this part, we analyze and taxonomize smartphones' deficits that can be alleviated by augmentation. Figure 2 depicts our devised taxonomy.

1) Processing Power: Processing deficiencies of mobile clients due to slow processing speed and limited RAM is one of the major challenges in mobile computing [69]. Mobile devices are expected to have high processing capabilities similar to computing capabilities of desktop machines for performing computing-intensive tasks to enrich user experience. Realizing such vision requires powerful processor being able to perform large number of transactions in a short course of time.

Large internal memory/RAM to store state stack of all running applications and background services is also lacking. Beside local memory limitations, memory leakage also intensifies memory restrains of mobile devices. Memory leakage is the state of memory cells being unnecessarily occupied by running applications and services or those cells that are not released after utilization. Garbage-collector-based languages like Java in Android[2] contribute to memory leakage due to failed or delayed removal of unused objects from the memory [70]. Android's kernel level transactions can also leak memory in the absence of memory management mechanisms [70], [71]. Moreover, inward deficiency and inefficient design and implementation of mobile applications can also waste scarce memory of mobile devices. Thus, in the absence of required memory, applications are frequently paused or terminated by the operating system leading to longer execution time, excessive resource dissipation, and ultimately mobile user experience degradation.

2) Energy Resources: Energy is the only non-replenishable resource in mobile devices that demands external resources to be replenished [72], [73]. Currently, energy requirement of a mobile device is supplied via lithium-ion battery that lasts only few hours if device is computationally engaged. Battery capacity is increasing at about 5 to 10% a year [74], [75] as battery cells are excessively dense [72]. Moreover, mobile device manufacturers endeavor to attain device lightness, compactness, and handiness, which prevent exploitation of bulky long-lasting batteries. User safety is another concern that confines manufacturers to produce low capacity batteries [76]. While explosion of a battery with few hundreds milliamperes capacity can jeopardize human life [77], explosion of a high-capacity battery can carry catastrophic consequences.

Energy harvesting efforts [78]–[80] seek to replenish energy from renewable resources, particularly human movement, solar energy, and wireless radiation, but these resources are mostly intermittent and not available on-demand [81]. For instance, a sitting mobile user at night cannot have any external power source in the absence of wall power and wireless radiations. Moreover, researchers aim at reducing the energy overhead in different aspects of computing, including hardware, OS, application, and networking interface [82], [83]. Efforts are directed to develop alternative energy resources such as nuclear batteries that will likely last months or years [84]. However, significant deal of R&D is needed to fulfill ever-increasing energy requirements of mobile users.

Hence, in the absence of long-spanning energy on mobile devices, alternative augmentation approaches play a vital role in maturing mobile and ubiquitous computing.

3) Local Storage: Drastic increasing in the number of applications and amount of digital contents such as pictures, songs, movies, and home films [85] from one hand and limited storage of mobile devices from the other hand decelerate usability of mobile devices. While PCs are able to locally store huge amount of data, smartphones are limited to few gigabytes of space which are mostly occupied by system files, user applications, and personal data. Therefore, frequent

[2]urlhttp://www.android.com/

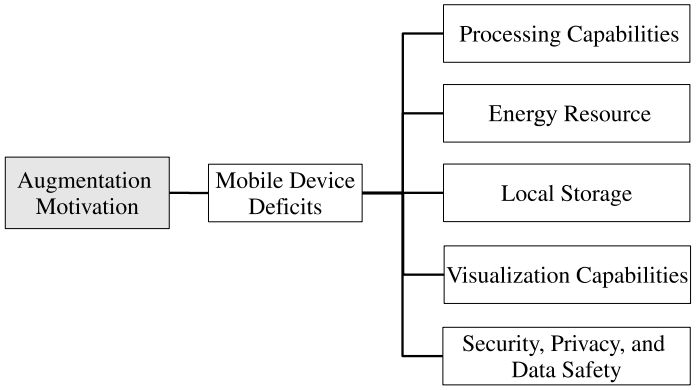

Fig. 2. Taxonomy of Augmentation Motivation: Intrinsic and non-intrinsic mobile challenges motivate augmentation.

storing, updating, and deleting data as well as uninstalling and reinstalling applications due to space limitation cause irksome impediments to mobile users [86]. Additionally, delivering offline usability, which is one of the most important characteristics of contemporary applications, remains an open challenge since mobile devices lack large local storage.

4) Visualization Capabilities: Effective data visualization on small mobile devices' screen is a non-trivial task when current screen manufacturing technologies and energy limitations do not allow significant size extensions without losing device handiness. Currently smartphones like HTC One X[3] and Samsung Galaxy Note II[4] have the biggest screens, at 4.7 and 5.5 inches respectively; however, they are very small compared to PCs and notebooks.

Therefore, efficient data visualization in small smartphones' screen necessitates software-based techniques similar to tabular pages, 3D objects, multiple desktops, switching between landscape and portrait views (needs accelerometer), and verbal communication to virtually increase presentation area. Recently, computing-intensive mobile 3D display technology is promising to noticeably mitigate the visualization deficit of contemporary smartphones. Glass-free auto-stereoscopic displays [87] can present 3D data by exploiting binocular parallax to offer a different view for each eye. Taking advantages of current and imminent software-based techniques beside native tools, especially tilting sensors significantly improve the mobile visualization capabilities in the near future. However, these approaches are computation-intensive processes that quickly drain battery [87], [88]. A feasible alternative solution to realize software-based content presentation approaches is to augment smartphones' computing capabilities.

5) Security, Privacy, and Data Safety: Mobile end-users are concerned about security and privacy of their personal data, banking records, and online behaviors [89]. The dramatic increase in cybercrime and security threats within mobile devices [90], cloud resources [91] and wireless transactions makes security and privacy more challenging than ever [92]. Moreover, performing complex cryptographic algorithms is likely infeasible because of computing deficiencies of mobile devices. Securing files using pair of credentials is also less realistic in the absence of large keyboard.

Data safety is another challenge of mobile devices, because information stored inside the local storage of mobile devices are susceptible to safety breaches due to high probability of hardware malfunction, physical damage, stealing, and loss.

Amalgam of these problems and deficiencies in mobile computing stimulates researchers from academia and industry to exploit novel technologies and approaches to augment computing capabilities of mobile devices which is subject of this study.

C. Mobile Augmentation Types: Taxonomy

In this Section, we analyze and classify augmentation approaches into two major types of hardware and software. Our devised taxonomy is depicted in Figure 3 and described as follows.

Hardware. The hardware approach aims to empower smartphones by exploiting powerful resources, particularly multi-core CPUs with high clock speed [93], large screen, and long-lasting battery [84], [94]. ARM[5] and Samsung[6] are major mobile processor manufacturers producing multi-core processors such as ARM Cortex-59[7] and Samsung Exynos 5 Octa core[8] that perform in higher speed than a single core processors [93]. However, doubling the CPU clock speed approximately octuples the device energy consumption [66].

Nevertheless, augmentation via sophisticated hardware is hindered by several obstacles. Firstly, generating powerful processor, large storage, and big screen decrease smartphone handiness due to additional heat, size, and weight. Secondly, considering the fact that utilizing long-lasting battery in small mobile devices is not feasible with current technologies, resource enlargement contributes toward faster battery drainage and shorter battery life time. Thirdly, equipping mobile devices with high-end hardware noticeably increases their price

[3]http://www.htc.com/www/smartphones/htc-one-x/

[4]http://www.samsung.com/my/consumer/mobile-devices/galaxy-note/galaxy-note/GT-N7100RWDXME

[5]http://arm.com

[6]http://samsung.com

[7]http://www.arm.com/products/processors/cortex-a50/cortex-a57-processor.php

[8]http://www.samsung.com/global/business/semiconductor/minisite/Exynos/blog_CES_2013_Samsung_Mobilizes_Possibility_with_Exynos5Octa.html

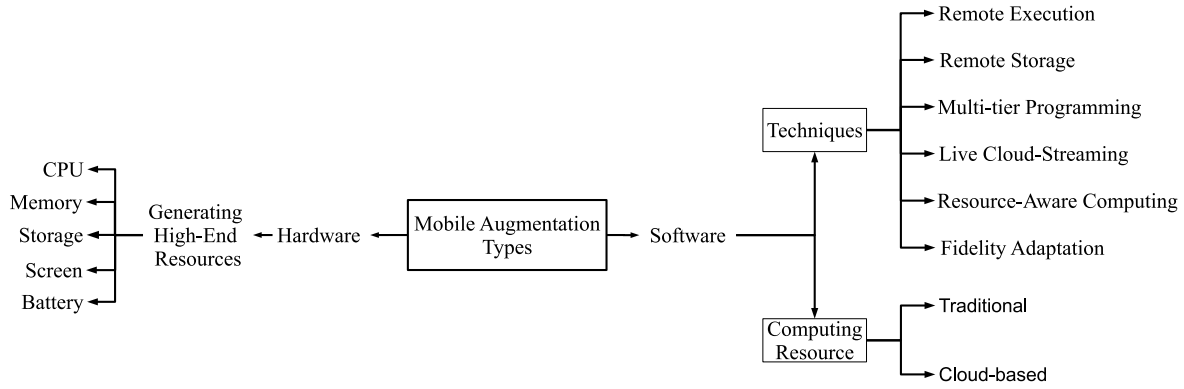

Fig. 3. Taxonomy of Mobile Augmentation Types.

compare to the stationary machines. Unlike PCs, smartphone's hardware is not upgradable; hence, a new device should be possessed in case of technology advancement. Therefore, in the absence of futuristic engineering technologies, the hardware-based augmentation process is slow and expensive that necessitates alternative augmentation approaches to enhance computing capabilities of mobile devices without drastic ownership price hike.

Software. Software-oriented mobile augmentation approaches are classified into five groups and will be explained later in this part. Resources that are used in major software-oriented approaches are classified into two groups, namely traditional and cloud-based. Their major differences lie on resource provisioning and access strategies, service security and delivery models, and resource characteristics. In traditional approaches, researchers leverage centralized resources of distant traditional servers or free nearby surrogates. Several problems such as resource availability, elasticity, and security of traditional approaches hinder their success. For instance, surrogates can terminate their services anytime without considering their current load, and can violate user security and privacy by changing execution sequence or altering raw and processed data.

To alleviate the problems of traditional servers, researchers in recent efforts [25], [27], [29], [31], [33]–[35], [41], [43], [44], [95] exploit highly available, elastic, secure cloud infrastructure. "Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers" [20].

While utilizing cloud resources, users pay for the amount and duration they utilize various resources (e.g., CPU, memory, and bandwidth) based on an agreed SLA. In the SLA, the amount and quality of required resources such as processor, RAM, and storage are specified and user is billed accordingly. Service delivery failure will be compensated by the vendor. Lucrative financial benefits of cloud services encourage cloud providers to compete in delivering high service availability, reliability, security, and robustness to increase their market share. Hence, the augmentation performance is less affected by resource unreliability and interruption.

Moreover, cloud infrastructures are available to end-users via Virtual Machine(VM)[9] to increase resource utilization ratio and enhance overall security and privacy. Virtualization technology aims to enable resource sharing in an isolated environment called VM. It realizes execution of multiple operating systems on a single machine and enables sharing of large resources among multiple end-users. Users can only access to infrastructures allocated to their VMs and cannot access prohibited resources and contents.

Table III summarizes the comparison results of traditional and cloud-based resources and advocates differentiations between the conventional servers and clouds. High computing power, elasticity, mobility support, low utilization overhead, and security are some of the significant advantages of cloud resources compare to the surrogates that advocate the latest paradigm shift in mobile augmentation.

Software augmentation techniques are classified as remote execution (offloading or cyber foraging) [5]–[8], [10]–[13], [16]–[18], [25], [29], [30], [33]–[35], [41], [43], [44], [96], remote storage [97], Multi-tier programming [36], [45], [46], live cloud-streaming [98], resource-aware computing [99], [100], and fidelity adaptation [101] and explained as follows.

• *Remote execution:* As explained in II-A,the resource-hungry components of mobile applications —in whole or part —are migrated to the resource-rich computing device(s) that are willing to share their resources with mobile devices. Rapid development of heterogeneous mobile devices obliges adaptive offloading approaches able to enhance capabilities of wide range of mobile devices in dynamic environment with least processing overhead and latency. The efficiency of offloading approaches highly depends on what component(s) can be partitioned? When partitioning takes place? Where to execute the component(s)? And how to communicate with the remote server? [102]. Offloading approaches perform varied-time analysis to answer these questions, which are classified into three groups and explained as below.

Design Time Analysis: In this method, the application's complexity is analyzed at design time to determine the answer of four above questions. Application developer or a middleware specifies the resource-intensive components of mobile

[9]http://www.vmware.com/virtualization/what-is-virtualization.html

TABLE III
COMPARISON BETWEEN TRADITIONAL AND CLOUD-BASED COMPUTING RESOURCE.

Features	Traditional	Cloud-Based
Computation Power	Low	High
Elasticity	Low	High
User Experience	Low	High
Reliability	Low	High
Availability	Intermittent	On-demand
Client Mobility	Limited	Unlimited
Multi-tenancy	Not available	Available
Serving Incentive	Not provisioned	Provisioned
Utilization Cost	Free	Pay-As-You-Use
Utilization Overhead	High	Low
Management	Decentralized	De/Centralized
Back-end Connectivity	Wired	Wired & Wireless
Communication Latency	Low	Varied
Computation Latency	High	Low
Security	Low	High
Data Safety	Low	High

application that can be offloaded to the remote server and label them as remote component(s). Programmers decide how to partition application and adapt its performance to the dynamic mobile environment which is a non-trivial task, mainly due to the lack of knowledge about the execution environment. Performing such action needs excessive programming skill and knowledge of computation offloading. Design time approaches [8], [10], [12] notably save native resources of mobile device by reducing the processing and monitoring overheads. However, partitioning prior to the execution is not always optimal and cannot accurately adapt performance in diverse execution environments and also imposes extra efforts on the application developer or middleware for deciding on partitioning. Hence, design time partitioning approaches are likely become obsolete.

Runtime Analysis: Runtime or dynamic partitioning referred to methods such as [25], [103] that aims to answer four questions at runtime. They identify and partition the resource-hungry parts of the application, specify how and where to execute the partitioned components [102], [104], and determine how to communicate with the server. In dynamic methods, resource requirement of the application is analyzed and available resources are detected to decide if the application requires remote resources. Upon decision making the system performs offloading. Further monitoring is necessary to gather knowledge of available remote resources to maintain execution history. Although these approaches provide dynamic and flexible solutions, large amount of resources are wasted at runtime that prolongs application execution and decrease energy efficiency.

Hybrid Analysis: The ultimate aim of hybrid approaches [105] is to increase performance and efficiency of augmentation methods. Deciding on how to perform the offloading mainly depends on the native resources, remote resources, and available network bandwidth. In [105], prior to the application execution, the system decides based on four options, namely i) no action, ii) dynamic, iii) static, and iv) profile only whether to offload or not and in case of offloading specify what type of partitioning should take place. The profile only option is similar to the no action, but the systems collect execution information to maintain execution history for future purpose.

• *Remote Storage:* Remote storage is the process of expanding storage capability of mobile devices using remote storage resources. It enables maintaining applications and data outside the mobile devices and provides remote access to them. In early efforts, researchers in [97] utilize iSCSI (Internet Small Computer System Interface) [106] —as a well-established protocol for remote storage —to access the server's I/O resources via mobile clients over the TCP/IP network to store, backup, and mirror data [107]. However, the throughput of iSCSI is highly affected by the mobile-server distance. Using iSCSI is also difficult for handling large files such as multimedia and database files. Moreover, due to message passing in wireless medium through TCP/IP, the security and processing overhead (e.g., cryptography and data compression) are further challenges. To alleviate these challenges, several researches as MiSC [108], UbiqStor [109], [110], and Intermediate Target [111] are proposed towards realizing remote storage on mobile devices. However, due to scalability, availability, performance, and efficiency issues of traditional servers, power of remote storage could not fully unleash using traditional servers.

Several proposals and data storage services in academia and industry aim to expand mobile storage by exploiting cloud computing, especially Jupiter [31], SmartBox [112], Amazon S3[10], Mozy[11], Google Docs[12], and DropBox[13]. For instance, Jupiter expands smartphone storage and assists end-users in organizing large applications and data. Jupiter lever-

[10]http://aws.amazon.com/s3/
[11]http://mozy.com
[12]https://docs.google.com/
[13]https://www.dropbox.com/

ages cloud infrastructures to store big data of mobile users. Heavy applications are executed inside the cloud's VM of smartphones and results are forwarded to the physical device after execution. Amazon S3 is a general purpose storage offering simple operations to store and retrieve cloud data while Mozy provides data backup facilities with main focus on enhancing cloud data safety against natural disasters.

• *Resource-Aware Computing:* In resource-aware computing efforts, especially [99], [100], [113]–[115], resource requirements of mobile applications are diminished utilizing the application-level resource management methods (using application management software such as compiler and OS) and lightweight protocols. Resource conservation is performed via efficient selection of available execution approaches and technologies [114]. Any mobile application consists of application-level resource management method is considered as a resource-aware application. For instance, in [100], authors propose an energy-friendly scheme for content-based image retrieval applications using three offloading options, namely i) local extraction-remote search, ii) remote extraction-remote search, and iii) remote extraction-local search. The authors consider available bandwidth, image database size, and number of user queries to opt any of three offloading options for saving energy. In a high bandwidth network with limited queries, the third option is beneficial; the system uploads all un-indexed images to the remote server and receives the results to be loaded into the memory. Then, all search queries are executed locally.

Similarly, applications can decide whether to choose 2G or 3G in telephony and FTP. Using 2G network for telephony and 3G for FTP can noticeably reduce resource requirements of the mobile applications, according to the power consumption patterns presented in [116]. 2G network technology consumes less energy for establishing a telephony communication, while 3G is more energy-friendly for file transfer transactions.

• *Fidelity adaptation:* Fidelity adaptation is an alternative solution to augment mobile devices in the absence of remote resources and online connectivity. In this method local resources are conserved by decreasing quality of application execution, which is unlikely desirable to end-users. As a well-known fidelity adaptation approach, we can refer to the YouTube[14]. Users in YouTube can adjust the streaming quality based on available bandwidth. To achieve optimized performance, researchers [78], [117] leverage composition of cyber foraging and fidelity adaptation.

• *Multi-tier Programming:* Developing distributed multi-tier mobile applications leveraging remote infrastructures is another technique employed in efforts such as [36], [45], [46], [118] to reduce resource requirements of mobile applications. The main idea in this type of mobile applications is to reduce the client-side computing workload and develop the applications with less native resource requirements. Certainly, the computationally intensive components of the applications are executed outside the device, whereas the interactive (user interface) and native codes (e.g., accessing to the device camera) remain inside the device for execution.

[14]http://youtube.com

Multi-tier applications are lightweight aiming to consume the least possible local resources by utilizing remote components and services, whereas native applications are monolithic applications often require runtime migration for execution. Therefore, monitoring time and communication overhead of multi-tier applications are shrunk leading to explicit resource saving and user experience enhancement.

• *Live Cloud Streaming:* In recent efforts to harness cloud resources, researchers from *Onlive*[15] and *Gaikai*[16], among other organizations introduce new approach to augment computing capabilities of mobile devices, entitled live cloud streaming [98]. In live cloud streaming approaches, mobile device acts as a dump client able to interact with server using a browser or application GUI. In live cloud streaming applications, entire processing take place in the cloud and results are streaming to the mobile devices. However, usability of cloud-streaming is hindered by latency, network bandwidth, portability, and network traffic cost.

Functionality of cloud-streaming applications absolutely depends on the network availability and the Internet. Transferring mobile-user input to the server is another critical factor that requires considerable attention under wireless Internet connection. Moreover, since majority of mobile network providers deploy 'pay-as-you-use' data plans, the large data traffic of cloud-streaming services imposes high communication cost on users. Yet congestion handling remains an open issue at peak hours. Entirely relying on cloud-streaming infrastructures and avoiding smartphones resources' utilization impact on application responsiveness and levy extravagant ownership, maintenance, power, and networking expenses to the cloud-streaming service providers, which is not a green computing approach.

III. Impacts of CMA on Mobile Computing

This Section discusses the advantages and disadvantages of performing a CMA process on mobile computing that are summarized in Table IV. We aim to demonstrate how CMA approaches mitigate deficiencies of mobile computing explained in Section II-B. In this Section the terms 'cloud resources' and 'cloud infrastructures' refer to any type of cloud-based resources and infrastructures discussed in Section V.

A. Advantages

In this part, eight major benefits of utilizing cloud resources in mobile augmentation processes are introduced.

1) Empowered Processing: Empowering processing is the state of virtually increased transaction execution per second and extended main memory leveraging CMA approaches. In computing-intensive mobile applications, either the hosting device does not have enough processing power and memory or cannot provide required energy. A common solution is to offload the application —in whole or part —to a reliable, powerful resource with least energy and time cost. In computation offloading, the complex, CPU- and memory-intensive

[15]http://onlive.com
[16]http://gaikai.com

components of a standalone application are migrated to the cloud. Consequently, the mobile devices can virtually perform and actually deliver the results of heavy transactions beyond their native capabilities.

Although surrogates in traditional augmentation approaches [8]–[10], [12] could increase computing capabilities of mobile devices, excessive overhead of arbitrary service interruption and denial could shadow augmentation benefits [19], [119]. Cloud resources guarantee highest possible resource availability and reliability.

Leveraging CMA approaches, application developers build mobile application with no consideration on available native resources of mobile devices and mobile users dismiss their devices' inabilities. Hence, computing- and memory-intensive mobile applications like content-based image retrieval applications (enable mobile users to retrieve an image from the database) can be executed on smartphones without excess efforts.

However, a flexible and generic CMA approach that can enhance plethora of mobile devices with least configuration, processing overhead, and latency is a vital need in excessively diverse mobile computing domain. Such diversity is mainly due to the rapid development of smartphones and Tablets, and sharp rise in their hardware, platform, API, feature, and network heterogeneity [120] in the absence of early standardization.

2) Prolonged Battery: Long-lasting battery can be considered as one the most significant achievements of CMA approaches for large number of mobile users. Smartphone manufacturers have already utilized high speed, multi-core ARM processors (e.g., Cortex-A57 Processor[17]) being able to perform daily computing needs of mobile end-users. However, such giant processing entities consume large energy and quickly drain the battery that irks end-users. CMA solutions can noticeably save energy [95] by migrating heavy and energy-intensive computing to the cloud for execution. Although energy efficiency is one of the most important challenges of current CMA systems, several efforts such as [53]–[55], [121], [122] are endeavoring to comprehend the energy implications of exploiting cloud-based resources from mobile devices and shrinking their energy overhead.

In traditional cyber foraging or surrogate computing approaches, energy is saved by computation offloading, but several issues such as lack of mobility support and resource elasticity can neutralize the benefits of energy-hungry task offloading.

3) Expanded Storage: Infinite cloud storage accessible from smartphones enables users to utilize large number of applications and digital data on device. Hence, they are not obliged to frequently install and remove popular applications and data due to the space limit. Online connectivity is essential to access cloud storage. In such online storage systems, data are manually or automatically updated to the online storage for maintaining the consistency of the online storage system. Storing applications in cloud storage provides the opportunity to update the code without consuming any mobile I/O transactions which enhances user experience and improves the smartphones' energy efficiency —because I/O transactions are energy-hungry tasks.

4) Increased Data Safety: CMA efforts can bring the benefit of data safety to the mobile users. Naturally, stored data on mobile devices are susceptible to loss, robbery, physical damage, and device malfunction. Storing sensitive and personal data such as online banking information, online credentials, and customer related information on such a risky storage significantly degrades the quality of user experience and hinders usability of mobile devices. Due to the scarce computing resources, especially energy in mobile devices, performing complex and secure encryption provisions is not feasible. Hence, by storing data in a reliable cloud storage [112], [123], users ensure data availability and safety regardless of time, place, and unforeseen mishaps. Threats such as device robbery or physical damage to the mobile devices will effect on the tangible value of the device rather than intangible value of the data.

5) Ubiquitous Data Access and Content Sharing: Cloud infrastructures play a vital role in enhancing data access quality. Storing data in cloud resources enables mobile users to access their digital contents anytime, anywhere, from any device. Hence, the impact of temporal, geographical, and physical differences is noticeably decreased that enriches user experience.

Moreover, cloud storages facilitate data sharing and contribution among authorized users. Every file and folder in cloud, usually has a protected unique access link that can be obtained by the owner to share them among legitimate users. Network traffic is hence, shrunk because data is accumulated in a central server accessible to unlimited users from various machines. Cloud can significantly enhance data transfer among different mobile devices. One of the most irksome user's impediments is to transfer data from current mobile device to a new handset. Apart from its temporal cost, porting data from one device to another, especially to a heterogeneous device is a risky practice that puts data is in the risk of corruption and loss of integrity. Stored data on Cloud remain safe and can be synchronized to any number of mobile devices with minimum risk. However, a reliable data access control mechanism is required to adjust user permissions.

6) Protected Offloaded Content: Cloud storage solutions aim to protect remote codes and data while ensure user's privacy. This is one of the most important gains of replacing surrogates with cloud resources. Cloud servers deploy virtualization technology to isolate the guest environment from other guests and also from their permanent software stack. Moreover, cloud vendors deploy strict security and privacy policies to not only ensure confidentiality of user content, but also to protect their properties and business. Implement internal security provisions particularly the state-of-the-art biometric security systems to protect their physical infrastructures and avoid unauthorized access. Employing complex content encryption, frequent patching, and continuous virus signature update inside the company premise or seeking technical services from a trusted third party [124] are other examples of security provisions undertaken in cloud to further protect cloud

[17]http://www.arm.com/products/processors/cortex-a50/cortex-a57-processor.php

TABLE IV
IMPACT OF CMA APPROACHES IN MOBILE COMPUTING.

Advantages	Disadvantages
Empowered Processing	Dependency to High Performance Networking Infrastructure
Prolonged Battery	Excessive Communication Overhead and Traffic
Expanded Storage	Unauthorized Access to offloaded Data
Increased Data Safety	Application Development Complexity
Ubiquitous Data Access and Content Sharing	Paid Infrastructures
Protected Offloaded Contents	Inconsistent Cloud Policies and Restrictions
Enriched User Interface	Service Negotiation and Control
Enhanced Application Generation	Nil

storage.

7) Enriched User Interface: As described in II-B4, visualization shortcomings of mobile devices diminish user experience and hinder smartphones' usability. However, cloud resources can be exploited to perform intensive 2D or 3D screen rendering. The final screen image can be prepared based on the smartphone screen size and streamed to the device. Consequently, screen adaptation also is achieved when cloud side processing engine automatically alter the presentation technique to match screen image with the device screen size.

8) Enhanced Application Generation: Cloud resources and cloud-based application development frameworks similar to μCloud and CMH, facilitate application generations in heterogeneous mobile environment. Once a cloud component is built, it can be utilized to develop various distributed mobile applications for large number of dissimilar mobile devices. In the presence of cloud components, application programmer needs to develop native mobile components and integrate them with relevant, prefabricated cloud components to develop a complex application. When a mobile-cloud application is developed for Android device, by slightly changing native components the application is transited to new OS like iOS[18] and Symbian[19] which significantly save time and money.

B. Disadvantages

Despite of many advantageous aspects of cloud services, their success is hindered by several drawbacks and shortcomings that are discussed as follows.

1) Dependency to High Performance Networking Infrastructure: CMA approaches demand converged wired and wireless networking infrastructures and technologies to fulfill intersystem communication requirements. In wireless domain, CMAs need high performance, robust, reliable, high bandwidth wireless communication to realize the vision of computing anywhere, anytime, from any-device. In wired communication, fast reliable communications ground is essential to facilitate live migration of heavy data and computations to a regional cloud-based resources near the mobile users. Efforts such as next generation wireless networks [125] and the open mobile infrastructure [126] with Open Wireless Architecture (OWA) by Sieneon [127] contribute toward enhancing the networking infrastructures' performance in MCC.

2) Excessive Communication Overhead and Traffic: Mobile data traffic is significantly growing by ever-increasing mobile user demands for exploiting cloud-based computational resources. Data storage/retrieval, application offloading, and live VM migration are example of CMA operations that drastically increase traffic leading to excessive congestion and packet loss. Thus, managing such overwhelming traffic and congestion via wireless medium becomes challenging, especially when offloading mobile data are distributed among helping nodes to commute to/from the cloud. Consequently, application functionality and performance decrease leading to user experience degradation.

3) Unauthorized Access to Offloaded Data: Since cloud clients have no control over their remote data, users contents are in risk of being accessed and altered by unauthorized parties. Migrating sensitive codes as well as financial and enterprise data to publicly accessible cloud resources decreases users privacy, especially enterprise users. Moreover, storing business data in the cloud is likely increasing the chance of leakage to the competitor firm. Hence, users, especially enterprise users hesitate to leverage cloud services to augment their smartphones.

4) Application Development Complexity: The excessive complexity created by the heterogeneous cloud environment increases environment's dynamism and complicates mobile application development. Mobile application developers are required to acquire extensive knowledge of cloud platforms (i.e., cloud OSs, programming languages, and data structures) to integrate cloud infrastructures to the plethora of mobile devices. Understanding and alleviating such complexity impose temporal and financial costs on application developers and decrease success of CMA-based mobile applications.

5) Paid Infrastructures: Unlike the free surrogate resources, utilizing cloud infrastructure levies financial charges to the end-users. Mobile users pay for consumed infrastructures according to the SLA negotiated with cloud vendor. In certain scenarios, users prefer local execution or application termination because of monetary cloud infrastructures cost. However, user payment is an incentive for cloud vendors to maintain their services and deliver reliable, robust, and secure services to the mobile users.

In addition, cloud vendors often charge mobile users twice;

[18]http://www.apple.com/ios/
[19]http://licensing.symbian.org/

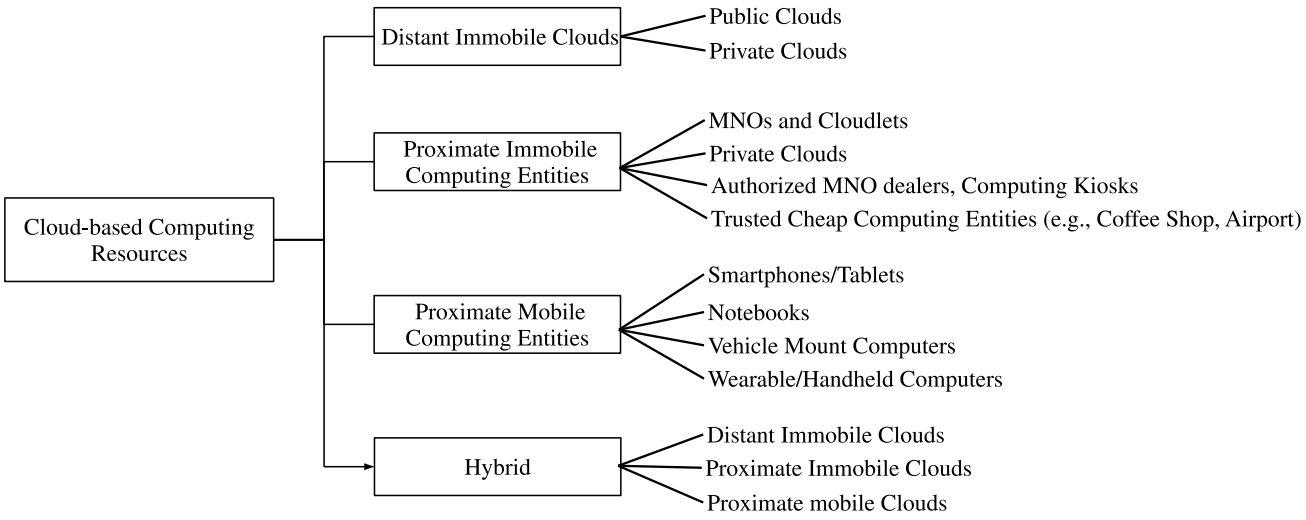

Fig. 4. Taxonomy of Cloud-based Computing Resources.

once for offloading contents to the cloud and once again when users decide to transfer their cloud data to another cloud vendors to utilize more appropriate service (e.g., monetary and QoS (Quality of Service) aspects).

6) Inconsistent Cloud Policies and Restrictions: One of the challenges in utilizing cloud resources for augmenting mobile devices is the possibility of changes in policies and restrictions imposed by the cloud vendors. Cloud service providers apply certain policies to restrain service quality to a desired level by applying specific limitations via their intermediate applications like Google App Engine bulk loader[20]. Services are controlled and balanced while accurate bills will be provided based on utilized resources.

Also, service provisioning, controlling, balancing, and billing are often matched with the requirements of desktop clients rather than mobile users [128]. Considering the great differences in wired and wireless communications, disregarding mobility and resource limitations of mobile users in design and maintenance of cloud can significantly impact on feasibility of CMA approaches. Hence, it is essential to amend restriction rules and policies to meet MCC users requirements and realize intense mobile computing on the go.

7) Service Negotiation and Control: While cloud users are required to negotiate and comply with the cloud terms and conditions for a certain period of time, often cloud agreements are nonnegotiable and policies might change over the time. Moreover, there is no control over the cloud performance and commitments in the absence of a controlling authority or a trusted third party. Hence, CMA services are always volatile to the service quality of cloud vendors.

IV. Taxonomy of Cloud-based Computing Resources

Researchers [24]–[27], [27]–[43], [45]–[49] aim to obtain user requirements and preferences by exploiting varied types of cloud-based resources to augment computing capabilities of resource-constraint smartphones. Based on the distance and mobility traits of such varied cloud-based computing resources, we classify them into four groups, namely distant immobile clouds, proximate immobile computing entities, proximate mobile computing entities, and hybrid that are taxonomized in Figure 4 and explained as follows. Table V represents the comparison results of these cloud-based computing resources. This Table can be utilized as a guideline for appropriate selection of cloud-based infrastructures in future CMA researches.

A. Distant Immobile Clouds

Public and private clouds comprised of large number of stationary servers located in vendors or enterprises premises are classified in this category. They are highly available, scalable, and elastic resources that are often located far from the mobile nodes accessible via the Internet. Although public cloud resources are likely more secure compared to the other types of resources due to complex security provisions and on-premise infrastructures [129]–[132], they are vulnerable to security attacks and breaches like Amazon EC2 crash [92] and Microsoft Azure security glitch [133]. Accessing cloud resources, especially public clouds often carries the risk of communicating through the risky channel of Internet. However, giant clouds are endeavoring to maintain security -for more market share- and could establish high reputation-based trust by providing long-term services to the users.

Additionally, the performance and efficacy of these approaches are affected by long WAN latency due to the long distance between mobile client and stationary cloud data centers. One potential approach to shorten the distance between mobile device and cloud is to migrate the remote code and data to the computing resources near to the mobile device via live migration of the VM from the cloud [134]. However, live migration of VM is a non-trivial task that requires great deal of research and development, particularly in networking environment due to several issues such as large VM size,

[20]https://developers.google.com/appengine/docs/python/tools/uploadingdata

TABLE V
COMPARISON OF CLOUD-BASED SERVERS.

	Distant clouds	Proximate immobile computing entities	Proximate mobile computing entities	Hybrid
Architecture	Distributed			
Ownership	Service provider	Public	Individual	Hybrid
Environment	Vendor Premise	Business Center	Urban Area	Hybrid
Availability	High	Medium	Medium	High
Scalability	High	Medium	Medium	High
Sensing Capabilities	Medium	Low	High	High
Utilization Cost	Pay-As-You-Use			
Computing Heterogeneity	High	Medium	High	High
Computing Flexibility	High	Medium	High	High
Power Efficiency	High	Medium	Medium	High
Execution Performance	High	Medium	Medium	High
Security and Trust	High	Moderate	Low	High
Utilization Rate	High			
Execution Platform	VM	VM	Physical/VM	Physical/VM
Resource Intensity	High	Moderate	Moderate	Rich
Complexity	Low	Moderate	Moderate	High
Communication Technology	3G/WiFi	WiFi	WiFi	3G/WiFi
Communication Latency	High	Low	Low	Moderate
Execution Latency	Low	Medium	Medium	Low
Maintenance Complexity	Low	Medium	Medium	High

hard-to-predict user mobility pattern, and limited, intermittent wireless bandwidth.

Resource utilization is enhanced in clouds due to the virtualization technology deployment. Several VMs can be executed on a single host to increase the utilization efficiency of the clouds, while each computation task runs on a single isolated VM loaded on a physical machine. However, VM security attacks such as VM hopping and VM escape [135] can violate the code and data security. VM hopping is a virtualization threat to exploit a VM as a client and attack other VM(s) on the same host. VM escape is the state of compromising the security of the hypervisor and control all the VMs.

B. Proximate Immobile Computing Entities

The second type of cloud-based computing resources involves stationary computers located in the public places near the mobile nodes. The number of computers in public places such as shopping malls, cinema halls, airports, and coffee shops is rapidly increasing. These machines are hardly performing tense computational tasks and are mostly playing music, showing advertisement, or performing lightweight applications. Moreover, they are connected to the power socket and wired Internet. Therefore, it is feasible to leverage such abundant resources in vicinity and perform extensive computation on behalf of resource-constraint mobile devices. It can also reduce latency and wireless network traffic while increases resource utilization toward green computing. Another group of proximate immobile computers are Mobile Network Operators (MNO) and their authorized dealers scattered in urban and rural areas, private clouds, and public computing kiosk [136] that can be exploited in smartphone augmentation.

However, protecting security and privacy of mobile user and computer owner hinder utilization of such nearby resources. Several shortcomings such as insufficient on-premise security infrastructure, lake of tight security mechanisms, and inefficient update and maintenance procedures inhibit utilizing such resources (except MNOs) for CMA approaches. Owners of these resources may attack mobile users and access their private data on the mobile devices or falsify offloading results. Also, malicious users may leverage these resources as an attacking point to violate mobile users' security and privacy. On the other hand, security and privacy of resource owners are also susceptible to violation. Owners of computer devices participating in resource sharing require robust mechanisms to protect and isolate the guest code and data from their host applications and data. Virtualization aims to realize such isolation mechanism, but issues such as VM hopping and VM escape require to be addressed before its successful adoption [135]. Among all proximate immobile resources, MNOs may be considered unique in terms of security and privacy features. MNOs, in general, have been serving mobile users for long time and could establish high degree of trust among mobile users. It is feasible to assume that MNO's certified dealers also can inherit MNO's trust if central management and monitoring process is undertaken by MNOs [46].

C. Proximate Mobile Computing Entities

In this category of cloud-based infrastructures, various mobile devices, particularly smartphones, Tablets, notebooks, wearable computers, and handheld computing devices play the role of servers based on cloud computing principles. The main benefit of utilizing nearby mobile resources is their

Fig. 5. The Hybrid Cloud Concept for MCC.

proximity to the mobile clients. Also, hardware and platform heterogeneity [51] between mobile servers and clients can be mitigated, because both sides are mainly ARM-based devices with mobile OSs. Moreover, contemporary smartphones are able to provide value added context- and social-aware services [137], [138] that contribute to the context-awareness of mobile applications. However, mobile devices' resources are limited and they are unable to perform intensive context-computing [139]. Realizing distributed computing on cluster of nearby mobile devices requires several issues, particularly application architectures, resource scheduling, and mobility to be addressed.

Moreover, security and privacy of mobile devices as a service provider is a critical concern in CMA. Mobile devices are intrinsically susceptible to loss and robbery, and their constraint resources inhibit exploiting robust security mechanisms inside the device. Furthermore, with ever-increasing popularity of mobile Apps (i.e., mobile applications) in online App stores such as Google Play[21] and Samsung APPs[22] [140] number of mobile security threats are rising sharply and malware-contaminated Apps are becoming serious threats to the mobile users [141]. Several security threats have been identified in an experiment of Android mobile applications with the potential to violate the security of mobile users [142]. Risk of such contaminated codes can likely be transferred to the non-contaminated mobile devices by utilizing their computation resources and request for results of a remote computation. Hence, establishing trust between mobile devices and end-users becomes a challenging task.

[21]https://play.google.com/store
[22]http://www.samsungapps.com

D. Hybrid (Converged Proximate and Distant Computing Entities)

Hybrid infrastructures as depicted in Figure 5 are comprised of various proximate and distant computing nodes, either mobile or immobile. The main idea behind building hybrid resources is to employ heterogeneous computing resources to create a balance between user requirements (mainly latency and computation power) and available options [143]. The latency sensitive codes are offloaded to the nearest computing device(s) whereas the most intensive and least latency sensitive tasks are migrated to the furthest resources. Perhaps, the utilization costs of nearby resources are more than the remote servers.

Beneficial characteristics of hybrid resources summarized in Table V advocates their usefulness in maximizing the augmentation benefits. However, deployment, management, and resource scheduling processes in dynamic mobile environment are non-trivial tasks. Developing an autonomic management system similar to CometCloud [144] in cloud computing and MAPCloud [143] in MCC to automatically manage, optimize, and adapt hybrid infrastructures in the cloud-mobile applications can significantly improve the quality of hybrid CMA approaches.

Hybrid cloud infrastructures can deliver enhanced security and privacy features to the CMA approaches and increase the QoS. Hybrid clouds are comprised of resources with varied security, privacy, and trust features which can be efficiently utilized by CMA and mobile users as a trade-off. For instance, security sensitive computations can perform a security-latency trade-off and execute computation inside a secure distant cloud.

V. The State-of-the-art CMA Approaches: Taxonomy

Cloud-based Mobile Augmentation (CMA) is the-state-of-the-art mobile augmentation model that leverages cloud com-

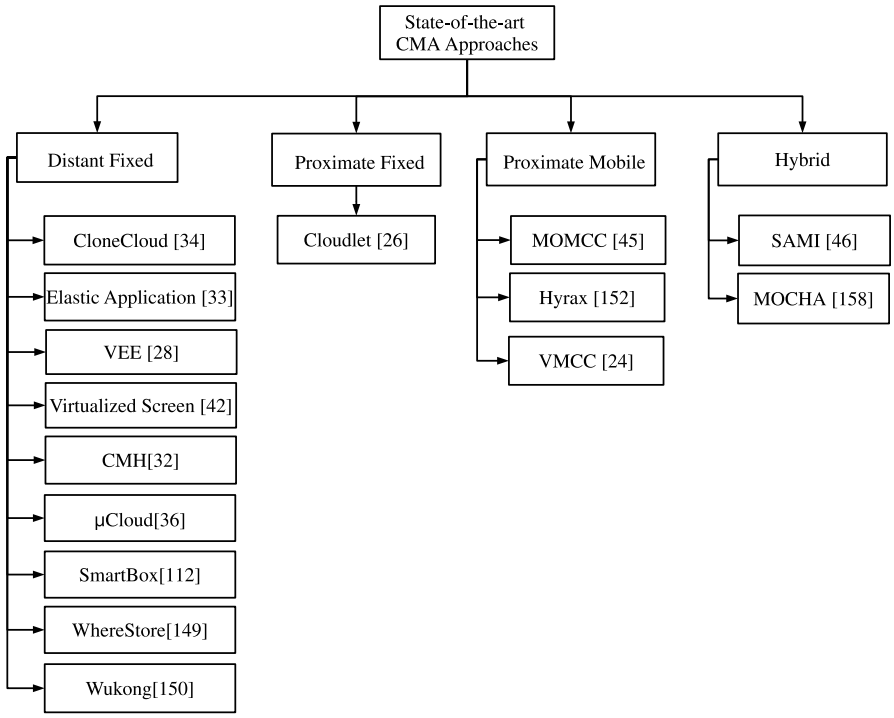

Fig. 6. Taxonomy of State-of-the-art CMA Models.

puting technologies and principles to increase, enhance, and optimize computing capabilities of mobile devices by executing resource-intensive mobile application components in the resource-rich cloud-based resources. According to our resource classifications in previous Section, we analyze and taxonomize the state-of-the-art CMA approaches into four models, namely distant fixed, proximate fixed, proximate mobile, and hybrid which are depicted in Figure 6. For each model, we describe few CMA efforts and tabulate the comparison results in Figure 10.

A. Distant Fixed

Majority of CMA approaches [25], [27], [29], [31], [33]–[35], [41], [43], [44], [54], [145] leverage fixed cloud infrastructures in distance due to its straightforward approach. Utilizing stationary cloud eliminates several management complexities (e.g., resource discovery and scheduling for mobile cloud-based servers) and alleviates reliability and security concerns [18]. Works in this class of CMA systems aim at reducing the complexity and overhead of utilizing distant cloud. For instance, in [54] authors propose an energy-efficient offline job scheduling model based on makespan minimization model to enhance energy efficiency of distant fixed CMA systems. Their main notion is to separate the data transmission from the job execution. During their work, authors provide several optimization solutions aiming to reduce the energy consumption of the device during the offloading process. However, for the sake of simplicity, the authors study the energy consumption of tasks in offline mode only which does not consider runtime dynamism of MCC.

Exploiting cloud resources is feasible in several real scenarios such as live cloud streaming [98], enterprise applications (e.g., Customer Relation Management (CRM) and enterprise resource planning [146]), and Social Networking. Cloud streaming mechanism has already described in II-C as an example of utilizing distance fixed resources. In [146], researchers leverage cloud resources in developing a CRM application to enhance efficiency of sale representatives for a pharmaceutical company. The representative meets the physician in medical centers to promote drugs, present samples and promotions material, and he records all sale results and details through the mobile application. The huge database of the company is stored inside the cloud and the sale representative can request to process, get, or update data in database without storing data locally.

We describe some of the distant fixed CMA approaches that utilize distant fixed cloud resources for mobile augmentation as follows. The terms immobile, fixed, and stationary are interchangeably used with the same notion.

- *CloneCloud:* CloneCloud [34] is a cloud-based, fine-grained, thread-level, application partitioner and execution runtime that clones entire mobile platform into the cloud VM and runs the mobile application inside the VM without performing any change in the application code. The CloneCloud enables local execution of remaining mobile application when remote server is running the intensive components unless local execution tries to accessing the shared memory state. Cloud resources in this effort simulate distributed execution of a monolithic application in a resourceful environment without engaging application developer into the distributed application programming domain. CloneCloud can significantly reduce the overall execution time using thread-level migration. When the local execution reaches the intensive compo-

nent(s), the CloneCloud system offloads the component(s) to the cloud and continues local execution until the application fetches data from the migrated state. The local execution is paused until the results are returned and integrated to the local application.

However, the communication overhead of transferring the clone of mobile platform, application, and memory state and frequent synchronization of the shared data between the mobile and cloud can shrink the power of cloud. Such overhead becomes more intense in case of heavy, data- and communication-intensive, and tightly coupled mobile applications where an alternative execution of resource-intensive and lightweight components exists. Frequent code and data encapsulation and migration, and mobile-cloud data synchronization excessively increase the communication traffic and impact on execution time and energy efficiency of the offloading.

- *Elastic Application:* Elastic application model [33] is a CMA proposal leverages distant fixed cloud data center for executing resource-intensive components of the mobile application. Authors in this model partition a mobile application into several small components, called weblet. Weblets are created with least dependency to each other to increase system robustness while decrease the communication overhead and latency. The weblet execution is dynamically configured to either perform locally or remotely, based on the weblet's resource intensity, execution environment quality, and offloading objectives. The distinctive attribute of this proposal is that application execution can be distributed among more than one machine and cooperative results can be pushed back to the device. To achieve such goal, multiple elasticity patterns namely replication, splitter, and aggregator are defined. In replication pattern, multiple replicas of a single interface are executed on multiple machines inside the cloud. Hence, failure in one replica will not compromise the system performance. In splitter pattern, the interface and implementation are separated so that several weblets with varied implementations can share a single interface. In aggregator, the results of multiple weblets are aggregated and pushed to the device for optimized accuracy and efficacy.

  The authors endeavor to specify the execution configurations (specifying where to run the weblets) at runtime to match the requirements of the applications and users. To enhance the overall execution performance and enrich user experience, the system is able to run the weblets both locally and remotely. A weblet can be executed remotely in a low-end device while the same can be executed locally on a high-end device.

  Elastic application model pays more attention to the user preferences by enabling different running modes of a single application (e.g., high speed, low cost, offline mode). However, it engages application developers to determine weblets organization based on the functionality, resource requirements, and data dependency. But, the characteristics of the weblets are mainly inherited from the well-known web services to decrease the programmer burdens.
- *Virtual Execution Environment(VEE):* Hung et al. [28] propose a cloud-based execution framework to offload and execute the intensive Android mobile applications inside the distant cloud's virtual execution environment. The quality and accuracy of execution environment is highly influenced by the comprehensiveness and accuracy of emulated platform. This method uses a software agent in both mobile and cloud sides to facilitate the overall system management. The agent in mobile device initiates VM creation and clones the entire application (even native codes and UI components) and partial data/memory state from device to the cloud. Unlike CloneCloud, VEE aims to reduce latency by migrating the segment of data stack explicitly created and owned by the application to the VM instead of copying the entire memory; cloning the entire memory state, especially for heavy applications significantly increases latency and traffic.

  During remote execution, the system frequently synchronizes the changes between device and cloud to keep both copies updated. In order to increase the quality and efficiency of remote execution in virtual environment and avoid data input loss at application suspension stages, the system stores input events (reading a file, capturing a face, storing a voice) exploiting a record/replay scheme and pseudo checkpoint methods. However, these methods engage application developers to separate the application state into two states, namely global and local and to specify the global data structures. The global state contains the program domain and application flow, whereas the local state contains local data structures required by a method. Programmer usually needs to identify global state when the application is paused. Once the application is suspended, the global state will be loaded to avoid re-execution and the latest Android checkpoint is applied to the system to reflect all the changes made from the last checkpoint. However, all changes, especially user input might be lost from the last checkpoint. To record the changes after the last checkpoint, the record/replay mechanism is deployed by creating a pseudo checkpoint. To create a pseudo checkpoint, the application notifies the local agent to identify the input events and record required information. Upon the application resumption, the pseudo checkpoint is restored to restore the application to the state prior to the suspension.

  In this effort, code security inside the cloud is enhanced by exploiting encryption and isolation approaches that protects offloaded code from cloud vendors eavesdropping. Using probabilistic communication QoS technique, this is aimed to provide a communication-QoS trade-off. For instance, the control data (usually small volume) needs highest accuracy while video streaming data (often large volume) requires less communication accuracy. Moreover, the authors are optimistic that offering secondary tasks such as automatic virus scanning, data backup, and file sharing in the virtual environment can enhance quality of user experience.

  Although this approach aims to enhance the quality of

application execution and augment computation capability of mobile clients and save energy, but responsiveness in interactive applications are likely low due to remote UI execution. Instead of migrating entire application to the cloud, it might be more beneficial to utilize some of the local mobile resources instead of treating mobile device as a dump client. Data passing between mobile device and cloud for interactive applications might degrade quality of experience, especially in low-bandwidth, intermittent networks.

- *Virtualized Screen:* Virtualized screen [42] is another example of CMA approaches that aims to move the screen rendering process to the cloud and deliver the rendered screen as an image to the mobile device. The authors aim to enrich the user experience and migrate the screen rendering tasks to the cloud with the assumption that majority of computation- and data-intensive processing take place in the cloud. Hence, abundant cloud resources' exploitation simplifies the CMA system architecture, prolongs mobile battery, and enhances the interaction and responsiveness of mobile applications toward rich user experience. Screen virtualization technique (running partial rendering in cloud and rest in mobile depending on the execution context) is envisioned to optimize user experience, especially for lightweight, high-fidelity, interactive mobile applications that entirely run on local resources. Their conceptual proposal aims to enhance visualization capability of mobile clients, mitigate the impact of hardware and platform heterogeneity, and facilitate porting mobile applications to various devices (e.g., smartphone, laptop, and IP TV) with different screens.
  To reduce the mobile-cloud data transmission, a frame-based representation system is exploited to forward the screen updates from the cloud to the mobile. Frame-based representation system captures and feeds the whole screen image to the transmission unit. This approach updates each frame based on the previous frame stored inside both the mobile and cloud. However, a rich interactive, responsive GUI needs live streaming of screen images which is impacted by communication latency. Although the authors describe optimized screen transmission approaches to reduce the traffic, the impact of computation and communication latency is not yet clear, as this is a preliminary proposal. Moreover, utilizing virtualized screen method for developing lightweight mobile-cloud application is a non-trivial task in the absence of its programming API.
- *Cloud-Mobile Hybrid (CMH) Application:* Unlike application offloading solutions, authors in this proposal [32] introduce a new approach of utilizing cloud resources for mobile users. In this effort, the authors propose a novel CMH application model, in which heavy components are developed for cloud-side execution, whereas lightweight or native codes are developed for mobile devices execution. CMH Applications execution does not need profiling, partitioning, and offloading processes and hence produce least computation overhead on mobile devices. Upon successful cloud-side execution, the results are returned back to the mobile for integrating to the native mobile components.
  However, developing CMH applications is significantly complex due to the interoperability and vendor lock-in problems in clouds and fragmentation issue in mobiles [51]. Cloud components designed for a specific cloud are not able to move to another cloud due to underlying heterogeneity among clouds. Similarly, mobile components developed for a particular platform cannot be ported to different platforms because of heterogeneity. Yet isolating development of mobile and cloud components creates further versioning and integration challenges.
  To mitigate the complexity of CMH application developments and facilitate portability, the authors leverage Domain Specific Language (DSL) [147], [148]. A DSL is a programming language with major focus on solving problem in specific domains. MATLAB[23] is a well-known DSL-based tool for mathematicians. A parser takes a DSL script and converts codes into an in-memory object to be forwarded to various automatic component generators. The system needs different code generators for various mobile and cloud platforms. Once the mobile and cloud components are generated, the CMH application can be assembled for various mobile-cloud platforms. However, utilizing DSL-based techniques requires more generalization efforts to be beneficial in developing all types of CMH applications.
- µ*Cloud:* Similar to the CMH framework, µCloud [36] is a modular, mobile-cloud application framework that aims to facilitate mobile-cloud application generation, promote application portability, minimize the development complexity, and enhance offline usability in intensive mobile-cloud applications. Fulfilling separation of concerns vision, skilled programmers independently develop self-contained components which do not have any direct inter communications with each other. Unskilled mobile users can mash-up (assembling available components to build complex application) these prefabricated components to generate a complex mobile-cloud application. Cloud vendors provide infrastructure and platform as cloud services to run prefabricated cloud components. The main idea in this proposal is to avoid local execution of the resource-intensive components. Hence, components are identified as cloud, mobile, and hybrid; mobile components are executable exclusively on mobile and cloud components are strictly developed for cloud server while hybrid components can either run locally or remotely. Hybrid components have either multiple implementations or a single implementation that need a middleware for execution. Each component has a triplet of identifier, input/output parameters, and configuration.
  To alleviate offline usability issue, the authors leverage mobile-side queuing and cloud-side caching to maintain data in case of disconnection. Data will be transferred upon reconnection. Application is partitioned into components and organized as a directed graph. Nodes

[23] http://www.mathworks.com/products/matlab/

represent components and vertices indicate data/control flow. Application is divided into three fragments; in each fragment, a managing unit called orchestrator executes and maintains component's mash-up process. The output of each component is forwarded using the pass-by-value semantic as an input to the subsequent component.
Unlike Elastic Application model, the design and implementation of components in µCloud is statically performed in early development phase. Thus, any improvement in resource availability of mobile devices or environmental enhancement (like bandwidth growth) will not improve the overall execution of µCloud applications. Such inflexibility decreases the application execution performance and degrades the quality of user experience.

- *SmartBox:* Smartbox [112] is a self-management, online, cloud-based storage and access management system developed for mobile devices to expand device storage and facilitate data access, and sharing. It is a write-once, read many times system designed to store personal data such as text, song, video, and movies which is not appropriate for large scale computational datasets. In Smartbox, mobile devices are associated with a shadow storage to store/retrieve personal data using a unique account. To facilitate data sharing among larger group of end-users in office or at home, a public storage space is provisioned.
Smartbox exploits traditional hierarchical namespace for smooth navigation and employs an attribute-based method to facilitate data navigation and service query using semantic metadata such as the publisher-provider metadata. Data navigation and query using tiny keyboard and small screen irk mobile users when inquiring and navigating stored data in cloud. However, mobile users need always-on connectivity to access online cloud data which is not yet achieved and is unlikely to become reality in near future.
- *WhereStore:* WhereStore [149] is a location-based data store for cloud-interacting mobile devices to replicate necessary cloud-stored mobile data on the phone. The main notion in this effort is that users in different places doing various activities need dissimilar types of information. For instance, a foreign tourist in Manhattan requires information about nearby places of interest rather than all the country. Hence, identifying the location and caching predicted data deemed can enhance the system efficiency and user experience. However, efficient prediction of future user location and required data, and determining the right time for caching data are challenging tasks.
- *Wukong:* Wukong [150] is a cloud oriented file service for multiple mobile devices as a user-friendly and highly available file service. The authors provision support of heterogeneous back-end services such as FTP, Mail, and Google Docs Service in a transparent manner leveraging a service abstraction layer (SAL). Wukong enables applications to access cloud data without being downloaded into the local storage of mobile device.
Authors introduce cache management and pre-fetch mechanisms in different scenarios to increase performance while decreasing latency. However, it cannot always reduce latency due to the bandwidth limitation and I/O overhead. In operations with long gap between open and read, it is beneficial to pre-fetch data from cloud to the device that significantly improves user experience. Data security is enhanced via an encryption module and bandwidth is saved using a compression module. While compression methods utilized in this proposal is inefficient for multimedia files like image and music, it can compress text and log files noticeably.
We conclude that one of the most effective solutions to tackle bandwidth and latency limitations in CMA approaches, especially cloud storage is to decrease the volume of data using imminent compression methods. While various compression methods work well on specific file types, a cognitive or adaptive compression method with focus on multimedia files can significantly improve the feasibility of cloud-storage systems.

B. Proximate Fixed

Researchers have recently proposed CMA approaches in which nearby stationary computers are utilized. Utilizing nearby desktop computers initiates new generation of services to the end-user via mobile device. In [26], the authors provide a real scenario in which Ron, a patient diagnosed with Alzheimer, receives cognitive assistance using an augmented-reality enabled wearable computer. The system consists of a lightweight wearable computer and a head-up display such as Google Glass[24] equipped with a camera to capture the environment and an earphone to send the feedback to the patient. The system captures the scene and sends the image to the nearby fix computers to interpret the scene in the image using the object or face recognition, voice synthesizer, and context-awareness algorithms. When Ron looks at a person for few seconds, the person's name and some clue information is whispered in Ron's ear to help greeting with the person. When he looks at his thirsty plant or hungry dog, the system reminds Ron to irrigate the plant and feed his dog. The nearby resources are core component of this system to provide low-latency real-time processing to the patient. In this part, we explain one of the most prominent proximate fixed efforts as follows.

- *Cloudlet:* Cloudlet [26] is a proximate immobile cloud consists of one or several resource-rich, multi-core, Gigabit Ethernet connected computer aiming to augment neighboring mobile devices while minimizing security risks, offloading distance (one-hop migration from mobile to Cloudlet), and communication latency. Mobile device plays the role of a thin client while the intensive computation is entirely migrated via Wi-Fi to the nearby Cloudlet. Although Cloudlet utilizes proximate resources, the distant fixed cloud infrastructures are also accessible in case of Cloudlet scarcity. The authors employ a decentralized, self-managed, widely-spread infrastructure built on hardware VM technology. Cloudlet is a VM-based

[24]http://www.google.com/glass/start/

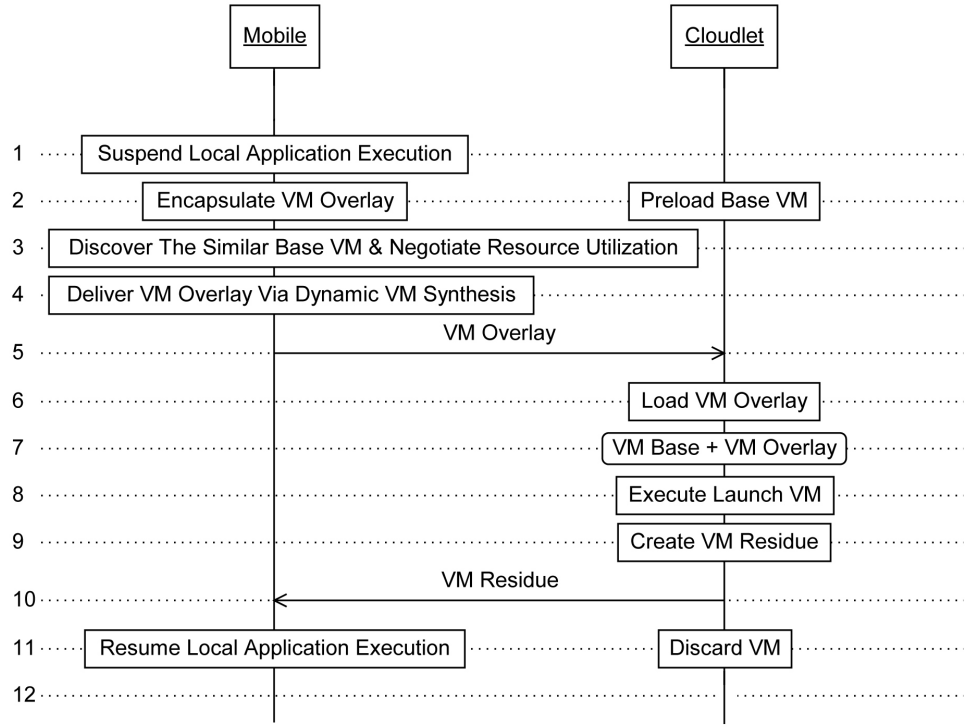

Fig. 7. Cloudlet-based Resource-Rich Mobile Computing Life Cycle.

offloading system that can significantly shrink the impact of hardware and OS heterogeneity between mobile and Cloudlet infrastructures.

To reduce the Cloudlet management and maintenance costs while increasing security and privacy of both Cloudlet host and mobile guest, a method called "transient Cloudlet customization" is deployed which uses hardware VM technology. It enables Cloudlet customization prior to the offloading and performs Cloudlet restoration as a post-offloading cleanup process to restore the host to its original software stake. The VM encapsulates the entire offloaded mobile environment (data state and code) and separates it from the host permanent software. Hence, feasibility of deploying Cloudlet in public places such as coffee shops, airport lounge, and shopping malls increases.

Unlike CloneCloud and Virtual Execution Environment efforts that migrate the entire mobile OS clone to the cloud, Cloudlet assumes that the entire OS clone exists and is preloaded in the host and runs on an isolated VM. In mobile side, instead of creating the VM of the entire mobile application and its memory stack, the systems encapsulates a lightweight software interface of the intensive components called VM overlay.

The VM overall offloading performance is further enhanced by exploiting Dynamic VM Synthesis (DVMS) method since its performance solely depends on the mobile-Cloudlet bandwidth and cloudlet resources. DVMS assumes that the base VM is already available in the target Cloudlet and user can find the matching execution environment (VM base) among silo of nearby Cloudlets. Upon discovery and negotiation of the Cloudlet, the DVMS offloads the VM overlay to the infrastructure to execute launch VM (base + overlay). Henceforth, the offloaded code starts execution in the state it was paused. Upon completion of Cloudlet execution the VM residue is created and sent back to the mobile device. In the Cloudlet, the VM is discarded as a post-offloading cleanup process to restore the original Cloudlet state. In mobile side, the results will be integrated to the application and local execution will be resumed. To present a clear understanding of the overall process, the sequence diagram of Cloudlet-based resource-rich mobile computing is depicted in Figure 7.

Despite the noticeable offloading improvements in the Cloudlet, its success highly depends on the existence of plethora of powerful Cloudlets containing popular mobile platforms' base VM. Encouraging individual owners to deploy such Cloudlets in the absence of monetary incentives is an issue that must be addressed before deployment in real scenarios. Although energy efficiency, security and privacy, and maintenance of Cloudlet are widely acceptable, further efforts are required to protect the overall CMA process. Moreover, few minutes offloading latency in Cloudlet is unacceptable to users [151].

C. Proximate Mobile

Recently, several researchers [24], [45], [122], [152]–[155] propose CMA approaches in which nearby mobile devices lend available resources to other mobile clients for execution

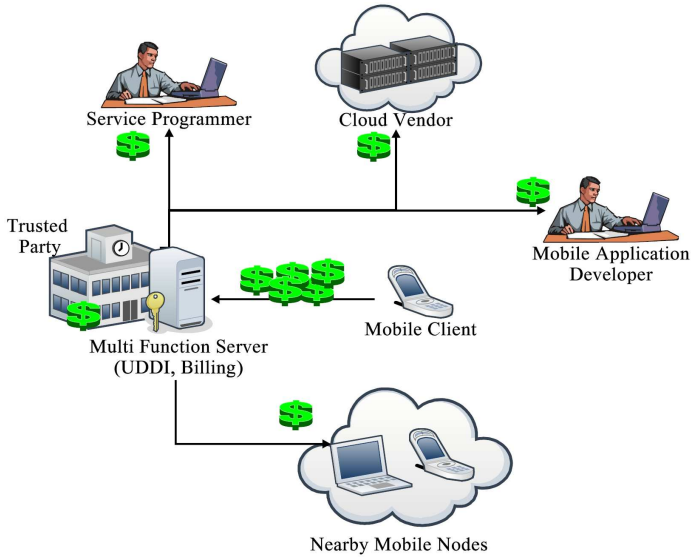

Fig. 8. MOMCC Concept.

of resource-intensive tasks in distributed manner. Utilizing such resources can enhance user experience in several real scenarios such as Optical Character Recognition (OCR) and natural language processing applications. The feasibility of utilizing nearby mobile devices is studied in [24] where Peter, a foreign tourist, visiting a Korean exhibition and finds interest in an exhibit, but cannot understand the Korean description. He can take a photo of the manuscript and translate it using the OCR application, but his device lacks enough computation resources. Although he can exploit the Internet web services to translate the text, the roaming cost is not affordable to him. Hence, he leverages a CMA solution by utilizing computation resources of nearby mobile devices to complete the task. Some of the CMA efforts whose remote resources are proximate mobile devices are explained as follows.

- *MOMCC:* Market-Oriented Mobile Cloud Computing (MOMCC) [45] is a mobile-cloud application framework based on Service Oriented Architecture (SOA) that harnesses a cluster of nearby mobile devices to run resource-intensive tasks. In MOMCC, mobile-cloud applications are developed using prefabricated building blocks called services developed by expert programmers. Service developers can independently develop various computation services and uploaded them to a publicly accessible UDDI (Universal Description Discovery and Integration) such as mobile network operators.
  Services are mostly executed on large number of smartphones in vicinity which can share their computation resources and earn some money. To enhance resource availability and elasticity, distant stationary cloud resources are also available if nearby resources are insufficient. In order to become an IaaS (Infrastructure as a Service) provider, mobile devices register with the UDDI and negotiate to host certain services after secure authentication and authorization. Mobile users at runtime contact UDDI to find appropriate secure host in vicinity to execute desired service on payment. The collected revenue is shared between service programmer, application developer, UDDI, and service host for promotion and encouragement. Figure 8 depicts the MOMCC concept.
  However, MOMCC is a preliminary study and its overall performance is not yet evident. Several issues are required to be addressed prior to its successful deployment in real scenarios. Executing services on mobile devices is a challenging task considering resource limitation, security, and mobility. Also an efficient business plan that can satisfy all engaging parties in MOMCC is lacking and demand future efforts. MOMCC can provide an income source for mobile owners who spend couple of hundred dollars to buy a high-end device. In addition, faulty resource-rich mobile devices that are able to function accurately can be utilized in MOMCC instead of being e-waste.
- *Hyrax:* Hyrax [152] is another CMA approach that exploits the resources from a cluster of immobile smartphones in vicinity to perform intense computations. Hyrax alleviates the frequent disconnections of mobile servers using fault tolerance mechanism of Hadoop. Similar to MOMCC and Cloudlet, due to resource limitations of smartphone servers, the accessibility to distant stationary clouds is also provisioned in case the nearby smartphone resources are not sufficient. However, Hyrax does not consider mobility of mobile clients and mobile servers. Hence, deployment of Hyrax in real scenarios may become less realistic due to immobilization of mobile nodes. Lack of incentive for mobile servers also hinders Hyrax success.
  Hyrax is a MCC platform developed based on Hadoop [156] for Android smartphones. In developing Hyrax, the MapReduce [157] principles are applied utilizing Hadoop as an open source implementation of MapReduce. MapReduce is a scalable, fault-tolerant programming model developed to process huge dataset over a cluster of resources. Centralized server in Hyrax runs two

client side processes of MapReduce, namely NameNode and JobTracker processes to coordinate the overall computation process on a cluster of smartphones. In smartphone side, two Hadoop processes, namely DataNode and TaskTracker are implemented as Android services to receive computation tasks from the JobTracker. Smartphones are able to communicate with the server and other smartphones via 802.11g technology.

Nevertheless, the cloud storage connectivity in Hyrax is missing. It demands several gigabytes of local storage to store data and computation. Hence, user cannot access distributed data over the Internet or Ethernet. The author utilizes the constant historical multimedia data to avoid file sharing. Hence, it is less beneficial for interactive and event-oriented applications whose data frequently changes over the execution and also data-intensive applications that require huge database. The overall overhead in Hyrax is high due to the intensity of Hadoop algorithm which runs locally on smartphones.

- *Virtual Mobile Cloud Computing (VMCC):* Researchers in [24] aim to augment computing capabilities of stable mobile devices by leveraging an ad-hoc cluster of nearby smartphones to perform intensive computing with minimum latency and network traffic while decreasing the impact of hardware and platform heterogeneity. During the first execution, required components (proxy creation and RPC support) are added to the application code to be used for offloading; the modified code will remain for future offloading. For every application, the system determines the number of required mobile servers, security and privacy requirements, and offloading overhead. The system continuously traces the number of total mobile servers and their geolocation to establish a peer-to-peer communication among them. Upon decision making the application is partitioned into small codes and transferred to the nearby mobile nodes for execution. The results will be reintegrated back upon completion.

  However, several issues encumber VMCC's success. Firstly, this solution, similar to Hyrax, is not suitable for a moving smartphone since the authors explicitly disregard mobility trait of mobile clients. Secondly, every computing job is sent to exactly one mobile node; so, the offloading time and overhead will be increased when the serving node leaves the cluster. Thirdly, the offloading initiation might take long since the offloading's overall performance highly depends on the number of available nearby nodes; insufficient number of mobile nodes defers offloading. Finally, in the absence of monetary incentive for mobile nodes the likelihood of resource sharing among resource-constraint mobile devices is low.

D. Hybrid

Hybrid CMA efforts are budding [46], [143], [158] to optimize the overall augmentation performance and researchers are endeavoring to seamlessly integrate various types of resources to deliver a smooth computing experience to mobile end-users. For instance, mCloud [159] is an imminent proposal to integrate proximate immobile and distant stationary computing resources. Authors are aiming to enable mobile-users to perform resource-intensive computation using hybrid resources (integrated cloudlet-cloud infrastructures). Hybrid solutions aim to provide higher QoS and richer interaction experience to the mobile end-users of real scenarios explained in previous parts. For instance, in the foreign tourist example, the image can be sent to the nearby mobile device of a non-native local resident for processing. When the processing fails due to lack of enough resources, the picture can be forwarded to the cloud without Peter pays high cost of international roaming (Peter may pay local charge).

We review some of the available hybrid CMAs as follows.

- *SAMI:* SAMI (Service-based Arbitrated Multi-tier Infrastructure for mobile cloud computing) [46] proposes a multi-tier IaaS to execute resource-intensive computations and store heavy data on behalf of resource-constraint smartphones. The hybrid cloud-based infrastructures of SAMI are combination of distant immobile clouds, nearby Mobile Network Operators (MNOs), and cluster of very close MNOs authorized dealers depicted in Figure 9. The compound three level infrastructures aim to increase the outsourcing flexibility, augmentation performance, and energy efficiency. The MNO's revenue is hiked in this proposal and energy dissipation is prevented. Nearby dealers can be reached by Wi-Fi. MNO's can be accessed either directly via cellular connection or through dealers via Wi-Fi and broadband. Connection is established via cellular network to contact distant stationary clouds. The cluster of nearby stationary machines (MNO dealers located in vicinity) performs latency-sensitive services and omits the impact of network heterogeneity. SAMI leverages Wi-Fi technology to conserve mobile energy because it consumes less energy compared to the cellular networks [116]. In case of nearby resource scarcity or end-user mobility, the service can be executed inside the MNO via cellular network. However, if the resources in MNO are insufficient, the computation can be performed inside the distant immobile cloud.

  The resource allocation to the services is undertaken by arbitrator entity based on several metrics, particularly resource requirements, latency, and security requirements of varied services. The arbitrator frequently checks and updates the service allocation decision to ensure high performance and avoids mismatch.

  To enhance security of infrastructures, SAMI employs comparatively reliable and trustworthy entities, namely clouds, MNOs, and MNO trusted dealers. MNOs have already established reputation-trust among mobile users and can enforce a strict security provisions to establish indirect trust between dealers and end users ensuring that user's security and privacy will not be violated. SAMI application development framework facilitates deployment of service-based platform-neutral mobile applications and eases data interoperability in MCC due to utilization of SOA.

  However, SAMI is a conceptual framework and deploy-

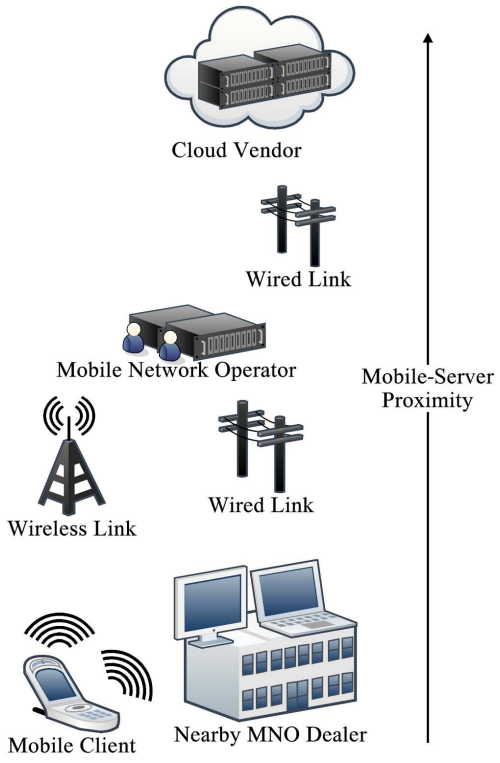

Fig. 9. SAMI: A Multi-Tier Cloud-based Infrastructure Model.

ment results are expected to advocate its feasibility. SAMI imposes a processing overhead on MNOs due to continuous arbitration process. Deployment, management, and maintenance costs of SAMI are also high due to the existence of various infrastructure layers. Moreover, though the authors discuss the monetary aspects of the proposal, a detailed discussion of the business plan is missing, for example in what scenario resource outsourcing is affordable for the mobile application? How does income should be divided among different entities to be satisfactory?

- *MOCHA:* In MOCHA [158] authors propose a mobile-cloudlet-cloud architecture for face recognition application using mobile camera and hybrid infrastructures of nearby Cloudlet and distant immobile cloud. Cloudlet is a specific, cheap cluster of computing entities like GPU (Graphics Processing Unit) capable of massively processing data and transactions in parallel. Cloudlets are able to be accessed via heterogeneous communication technologies such as Wi-Fi, Bluetooth, and cellular. The mobile often access processing resources via Cloudlet rather than directly connecting to the cloud, unless accessing cloud resources bears lower latency.
  Cloudlet receives the smartphones intensive computation tasks and partitions them for distribution between itself and distant immobile clouds to enhance QoS [26]. MOCHA leverages two partitioning algorithms: fixed and greedy. In the fixed algorithm, the task is equally partitioned and distributed among all available computing devices (including Cloudlet and cloud servers), whereas in greedy algorithm, the task is partitioned and distributed among computing devices based on their response times; the first partition is sent to the quickest device while the last partition is sent to the slowest device. The response time of the task partitioned using greedy approach is significantly better than fixed, especially when Cloudlet server is utilized in augmentation process and large number of clouds with heterogeneous response time exist. However, smartphones in MOCHA require prior knowledge of the communication and computation latency of all available computing entities (Cloudlet and all available distant fixed clouds) which is a resource-hungry and time-consuming task.

VI. CMA Prospectives

People dependency to mobile devices is rapidly increasing [89], [160] and smartphones have been using in several crucial areas, particularly healthcare (tele-surgery), emergency and disaster recovery (remote monitoring and sensing), and crowd management to benefit mankind [161]–[163]. However, intrinsic mobile resources and current augmentation approaches are not matching with the current computing needs of mobile-users, and hence, inhibit smartphone's adoption. Upon slow progress of hardware augmentation, the highly feasible solution to fulfill people computing needs is to leverage CMA concept. This Section aims to present set of guidelines for efficiency, adaptability, and performance of forthcoming CMA solutions. We identify and explain the vital decision making factors that significantly enhance quality and adaptability of future CMA solutions and describe five major performance limitation factors. We illustrate an exemplary decision making flowchart of next generation CMA approaches.

A. CMA Decision Making Factors

These factors can be used to decide whether to perform CMA or not and are needed at design and implementation phases of next generation CMA approaches. We categorize the factors into five main groups of mobile devices, contents, augmentation environment, user preferences and requirements, and cloud servers, which are depicted in Figure 11 and explained as follows.

1) Mobile Devices: From the client perspective, amount of native resources including CPU, memory, and storage is the most important factor to perform augmentation. Also, energy is considered a critical resource in the absence of long-spanning batteries. The trade-off between energy consumed by augmentation and energy squandered by communication is a vital proportion in CMA approaches [73]. Device mobility and communication ability (supporting varied technologies such as 2G,3G,Wi-Fi) are other metrics that are important in the offloading performance.

2) Contents: Another influential factor for CMA decision making is the contents' nature. The code granularity and size as well as data type and volume are example attributes of contents that highly impact on the overall augmentation process. Hence, the augmentation should be performed considering the nature and complexity of application and data.

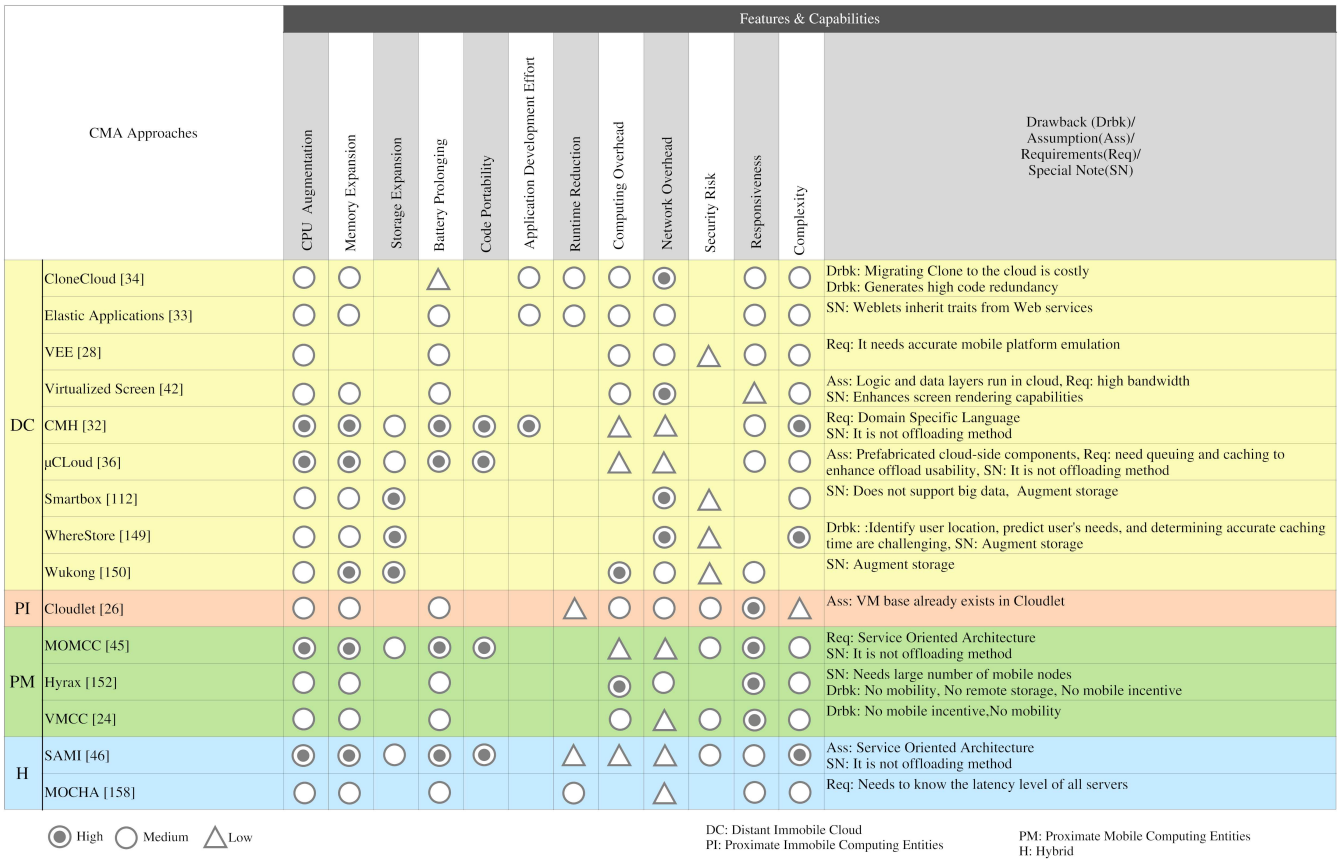

Fig. 10. Comparison of CMA Approaches.

For instance, latency sensitive small data are efficient to be processed locally, whereas sensitive big data are encouraged to be stored in a large reliable cloud storage. Similarly, offloading a coarse-grained, large code to a distant fixed cloud via a low bandwidth network is not feasible.

3) Augmentation Environment: Mobile computing is a heterogeneous environment comprised of non-uniform mobile nodes, communication technologies, and resources. One of the most influential environment-dependent factors is the wireless communication medium in which majority of communications take place. Wireless is an intermittent, unreliable, risky, and blipping medium with significant impact on the quality of augmentation solutions. The overall performance of a low cost, highly available, and scalable CMA approach is magnificently shrunk by the low quality of communication medium and technologies. Selecting the most suitable technology considering the factors like required bandwidth, congestion, utilization costs, and latency [164] is a challenge that affects quality of augmentation approaches in wireless domains. Wireless medium characteristics impose restrictions when specifying remote servers at design time and runtime.

Moreover, dynamism and rapidly changing attributes of the runtime environment noticeably impact on augmentation process and increase decision making complexity. Augmentation approaches should be agile in dynamic mobile environment and instantaneously reflect to any change. For example, user movement from high bandwidth to a low bandwidth network, receding from the network access point, and rapidly changing available computing resources complicate CMA process.

4) User Preferences and Requirements: End-users' physical and mental situations, individual and corporate preferences, and ultimate computing goals are important factors that affect offloading performance. Some users are not interested to utilize the risky channel of Internet, while others may demand accessing cloud services through the Internet. Hence, users should be able to modify technical and non-technical specifications of the CMA system and customize it according to their needs. For example, user should be able to alter degree of acceptable latency against energy efficiency of an application execution. Selecting the most appropriate resource among available options can also enhance overall user experience.

5) Cloud Servers: As explained, CMA approaches can leverage various types of cloud resources to enhance computing capabilities of mobile devices. Therefore, the overall performance and credibility of the augmentation approaches highly depend on the cloud-based resources' characteristics. Performance, availability, elasticity, vulnerability to security attacks, reliability (delivering accurate services based on agreed terms and conditions), cost, and distance are major characteristics of the cloud service providers used for augmenting mobile devices.

Utilizing clouds to augment mobile devices notably reduces the device ownership cost by borrowing computing resources based on pay-as-you-use principle. Such elastic, cost-effective,

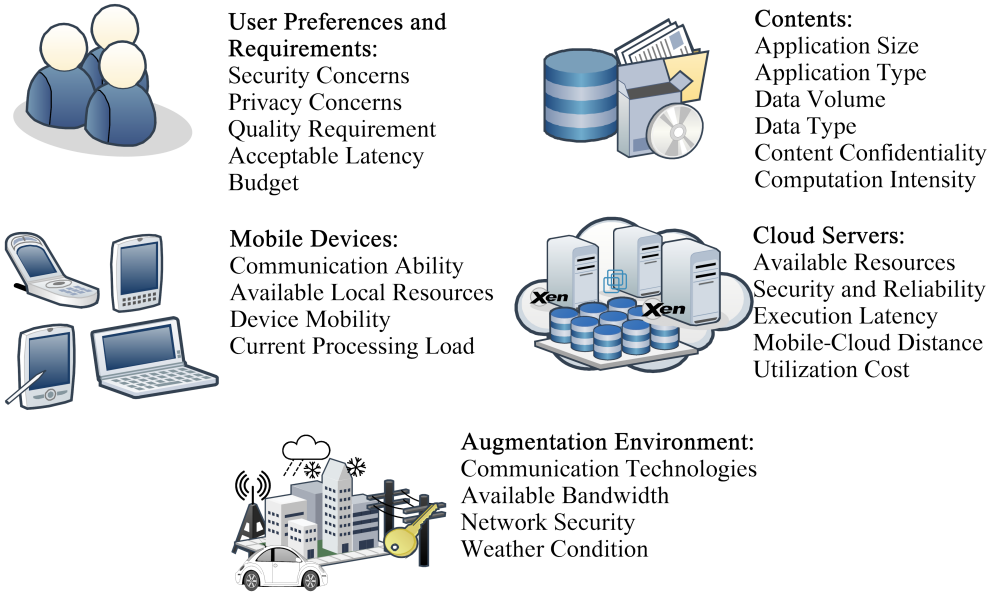

Fig. 11. Critical Factors in CMA Decision Making.

reliable, and relatively trustworthy resources are embraced by the scholars, industrial organizations, and end-users towards flourishing CMA approaches.

B. Performance Limitation Factors

Performance of varied CMA solutions is impacted by several factors. We describe fix major performance limitation factors as follows.

1) Heterogeneity: MCC is a highly heterogeneous environment comprised of three diversified domains of mobile computing, cloud computing, and networking. Although heterogeneity can provide flexibility to the mobile users by providing selection alternatives, it breeds several limitations and challenges, especially for developing multi-tier CMA-based applications [51]. Dissimilar mobile platforms such as Android, iOS, Symbian, and RIM beside diverse hardware characteristics of mobile device inhibit data and application portability among varied mobile devices. Portability is the ability to migrate code and data from one device to another with no/less modification and change [165]. Existing heterogeneity in cloud computing including hardware, platform, cloud service policy, and service heterogeneity originates challenges such as portability and interoperability and fragment the MCC domain.

Network heterogeneity in MCC is the composition of various wireless technologies such as Wi-Fi, 3G, and WiMAX. Mobility among varied network environments intensifies communication deficiencies and stems complex issues like signal handover [125]. Inappropriate decision making during the handover process like (i) less appropriate selection of network technology among available candidates and (ii) transferring the communication link at the wrong time, increases WAN latency and jitter that degrade quality of mobile cloud services. Consequently delay-sensitive content and services are degraded [166] and adoption of CMA approaches are hindered.

2) Data Volume: Ever-increasing volume of digital contents [85] significantly impacts on the performance of CMA approaches in MCC. Current wireless infrastructures and technologies fail to efficiently fulfill the networking requirements of CMA approaches. Storing such a huge data in a single warehouse is often impossible and demands data partitioning and distributed storage that not only mitigates data integrity and consistency, but also makes data management a pivotal need in MCC [167]. Applying a single access control mechanism for relevant data in various storage environments is another challenging task that impacts on the performance and adoption of CMA solutions in MCC.

3) Round-Trip Latency: Communication and computation latency is one of the most important performance metrics of mobile augmentation approaches, especially when exploiting distant cloud resources. In cellular communications, distance from the base station (near or far) and variations in bandwidth and speed of various wireless technologies affect the performance of augmentation process for mobile devices. Moreover, leveraging wireless Internet networks to offload content to the distant cloud resources creates a bottleneck. Latency adversely impacts on the energy efficiency [73] and interactive response [168] of CMA-based mobile applications due to excessive consumption of mobile resources and raising transmission delays.

Recently, researches [169], [170] are emerging toward decreasing the networking overhead and facilitating mobility (both node and code mobility) in cloud-based offloading approaches. For example, Follow-Me Cloud [169] aims at enabling mobility of network end-points across different IP

subnets. The authors employ the concept of identifier and locator separation of edge networks using OpenFlow-enabled switches. Leveraging the Follow-Me Cloud, mobile nodes can move among access networks without being notified of any change or session disruption. All corresponding nodes that have been communicating with the mobile node can continue their communication without interruption. When the node migrates, its old IP turn to identifier and its new IP address becomes locator address so that all other nodes can keep communication with the moving node. However, for each packet traveling to/from the mobile node, there is an overhead of manipulating the locator/identifier values. Future improvement and optimization efforts will enhance the CMA systems' performance.

In cloud side, computation latency significantly impacts on the application responsiveness. Researchers study the impact of cloud computation performance on the execution time and vindicate 12X reduction in performance time violation [171]. Thus, the increased latency degrades the quality of user experience and adversely impacts on the user-perceived performance of CMA solutions.

4) Context Management and Processing: Performance of CMA approaches is noticeably degraded by lack of sufficient, accurate knowledge about the runtime environment. Contemporary mobile devices are capable of gathering extensive context and social information such as available remote resources, network bandwidth, weather conditions, and users' voice and gestures from their surrounding environment [137], [138]. But, storing, managing, and processing large volume of context information (considering MCC environment's dynamism and mobile devices' mobility) on resource-constraint smartphones are non-trivial tasks.

5) Service Execution and Delivery: SLA as a formal contract between service consumer and provider enforces resource-level QoS (e.g., memory capacity, compute unit, and storage) against a fee, which is not sufficient for mobile users in highly dynamic wireless MCC environment. User-perceived performance in MCC is highly affected by the quality of cloud computations, wireless communications, and local execution. Hence, varied service providers, including cloud vendors, wireless network providers, and mobile hardware and OS vendors need to collaborate and ensure acceptable level of QoS. For successful CMA approaches, comprehensive real-time monitoring process is expected to ensure that engaging service vendors are delivering required services in acceptable level based on the accepted SLA.

C. CMA Feasibility

Although CMA is beneficial and can saves resources [40], several questions need to be addressed before CMA can be implemented in real scenarios. For instance: is CMA always feasible and beneficial? Can CMA save local resources and enhance user experience? What kind of cloud-based resources should be opted to achieve the superior performance?

Vision of future CMA proposals will be realized by accurate sensing and acquiring precise knowledge of decision making factors like user preferences and requirements, augmentation environment, and mobile devices, which are explained in previous part. A decision making system, similar to those used in [25], [33], [49], analyzes these vital factors to determine the augmentation feasibility and specifies if augmentation can fulfill mobile computation requirements and enrich quality of user experience. Figure 12 illustrates a possible decision making flow of future CMA approaches.

Availability of mobile resources to manage augmentation process and volume of cloud resources to provision required resources significantly impact on the quality of augmentation [9]. Similarly, user preferences, limitations, and requirements affect the augmentation decision making. For instance, if augmentation is not permitted by users, the application execution and data storage should be performed locally without being offloaded to a remote server(s) or be terminated in the absence of enough local resources. Similarly, augmentation process can be terminated if the execution latency of delay-sensitive content is sharply increased, quality of execution is noticeably decreased, or security and privacy of users is violated [19].

Furthermore, usefulness of CMA approaches highly depends on the execution environment. Offloading computation and mobile-cloud communication ratio, distance from mobile to the cloud, network technologies and coverage, available bandwidth, traffic congestion, deployment cost, and even nature of augmentation tasks alter usefulness of the CMA approaches [40]. For instance, performing an offloading method on a data-intensive application (e.g., applying a graphical filter on large number of high quality images) in a low-bandwidth network imposes large latency and significantly degrades user experience which should be avoided. Similarly, migrating a resource-hungry code to an expensive remote resource can be unaffordable practice. Suppose in a sample augmentation approach R_C is the total native resources consumed during augmentation, R_M is the total native resources consumed for maintenance, and R_S is the total resources conserved in augmentation process. Explicitly for a feasible augmentation approach $R_C + R_M << R_S$. However, in some traditional augmentation approaches, the left side of the equation exceeds the right which is not effective in augmenting resource-poor mobile devices [19].

VII. Open Challenges

In this Section, we highlight some of the most important challenges in deploying and utilizing CMA approaches as the future research directions.

A. Reference Architecture for CMA Development

Recently, researchers leverage dissimilar structures and techniques in utilizing cloud resources to augment computation capabilities of mobile devices. Also, different efforts focus on varied types of mobile applications, particularly multimedia, intense games, image processing, and workflow processing applications.

Such diffusion scatters CMA development approaches and increase adaptability challenges of CMA solutions. In the absence of a reference architecture and unified CMA solution, various CMA approaches need to be integrated to all mobile

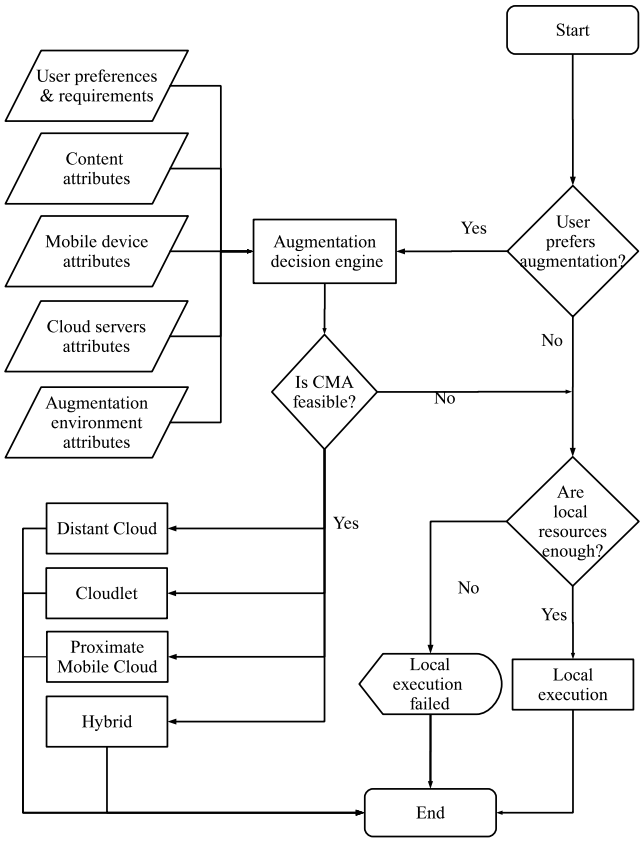

Fig. 12. An Exemplary Decision Making Flow of Future CMA Approaches.

OSs to serve multi-dimensional needs of various mobile users which is a not-trivial task. The reference architecture is expected to be generic enough to be deployed in family of CMA approaches.

B. Autonomic CMA

CMA approaches are drastically increasing volume of distributed mobile content in a horizontally heterogeneous mobile cloud computing environment [120]. Exploiting heterogeneous communication technologies to employ diverse cloud-based infrastructures for augmenting a plethora of dissimilar mobile devices is significantly intensifying complexity and management. Employing lateral solutions and controlling the mobile phones using outside entities (e.g., third party management systems) to deal with such complexity might further amplify complexity. A feasible alternative to mitigate such complexity is to develop autonomic self-managing, -healing, -optimizing, and -protecting CMA approaches [172] able to adapt to environment dynamism and hide complexity.

C. Application Mobility Provisioning

Enabling continuous and consistent mobility in CMA models (especially proximate mobile and hybrid) to provision ubiquitous, convenient, on-demand network access to cloud-based computing resources is a vital challenge. Seamless code mobility in CMA models is more challenging compared to the traditional augmentation approaches, because in CMA approaches service providers and consumers can move during the augmentation process which intensifies the code mobility [173]. Therefore, communication disruption and intermittency can cause several challenges, especially dismissal of always-on connectivity, excessive consumption of limited mobile resources, and frequent interruption of application execution which decreases quality of computing services and degrades quality of mobile user experience [72]. Also, it levies redundant costs on cloud-mobile users and inhibits reliability of CMA models. Hence, alleviating such difficulties using Web advancements [170] and imminent lightweight cognitive mobility management systems with least signal traffic and latency can significantly enhance the ubiquitous connectivity and increase the positive impact of CMA.

D. Computing and Temporal Cost of Mobile Distributed Execution

Noticeable computation and communication cost of migrating tasks from the mobile device to the remote servers and receiving the results is another challenges of CMA approaches in MCC, which is intensified by mobility and wireless communication constraints. Although researchers [45], [152] endeavor to reduce the distance of mobile devices and service providers by leveraging nearby mobile/fixed computing devices, several mechanisms, particularly resource discovery and allocation, service consumer and provider mobility management, and distributed runtime are required to realize the CMAs vision. Accurate allocation of resources to the mobile computation tasks demands comprehensive knowledge about structure and performance features of available service providers and resource requirements of mobile computation tasks. Thus, QoS-aware scheduling efforts such as [143] are necessary to enhance the CMA usability.

E. Seamless Communication

Maintaining a continuous communication between mobile service consumers and mobile/fixed service providers in intermittent heterogeneous wireless environment is a non-trivial task. User mobility and wireless disconnection highly impact on resource utilization ratio. When the mobile service providers and consumers loss communication link due to mobility-made prolonged distance, the service consumer requires to either performing local execution or re-initiating augmentation process. Similarly, the resource-constraint mobile server consumes its scarce resources for processing an orphan computation whose results are ineffectual after losing the communication link. Potential solutions may transfer partially-completed tasks to a nearby node or initiate parallel execution on third device before disconnection, or cache results for future references.

F. Multipoint Data Bridging

Unlike traditional offloading methods that require point-to-point code and data migration and processing, CMA approaches require multipoint data migration and interoperation

to achieve the maximum benefits from the distributed heterogeneous infrastructures in mobile-cloud ecosystem. Connecting heterogeneous systems (based on wired or wireless), understanding geographical information resources, and exchanging data between/across two or more heterogeneous systems [174] are the main issues in CMA system which demand arousing data interoperation techniques in multi-domains mobile cloud environment. The inward heterogeneous architectures and data structures of mobile devices and cloud systems with different APIs can exemplify the intensity of multipoint data bridging challenge [23]. Hence, offloading computational tasks from a mobile device to a cloud, performing computational interoperation among varied clouds (for cost and performance concerns), and pushing results to the mobile device become challenging tasks [175]. Therefore, multipoint data bridging in dynamic heterogeneous environment remains as a future research direction to realize accessing, interpreting, processing, sharing, and synchronizing distributed contents.

G. Distributed Content Management

Rapid growth in digital contents and increasing dependency of mobile users to cloud infrastructures impede content management for mobile users. Researchers distribute code and data among heterogeneous nearby and distant resources via different communication technologies to optimize CMA process. Although executing complex, heavy applications and accessing large data volume are facilitated, managing huge volume of distributed contents for smartphone users is not straightforward. Therefore, enabling mobile users to efficiently locate, access, update, and synchronize highly distributed contents requires future research and developments.

H. Seamless/Lightweight CMA

Developing lightweight mobile computing augmentation approaches to increase quality of mobile user experience and to develop CMA system independent of any particular situation is a significant challenge in mobile cloud environment. Offloading bulk data in limited wireless bandwidth, and VM initiation, migration, and management in a secure and confidential manner are particular tasks in CMA system that noticeably increase overall execution time, intensify the augmentation latency, and decrease the quality of mobile user experience. CMA approaches are generally hosted and executed inside the mobile devices to conserve their local resources and hence, need to avoid excessive resource hungry transactions.

A feasible approach to decrease the volume of digital contents —in limited bandwidth networks —is to utilize effective, efficient data compression methods. Available compression techniques are unlikely efficient considering structure of current multimedia files. Moreover, approaches like Paravirtualiztion [176] as a lightweight virtualization technique can reduce the overhead by partially emulating the OS and hardware. Paravirtualization approaches virtualize only parts of the hardware required for computing. Thus, the mobile-side VM creation overhead is diminished and the impact of VM migration on network is reduced. Therefore, realizing lightweight CMA approaches demands lightweight computation and communication techniques (particularly in virtualization, data compression, and encryption methods) to reduce intra-system correspondence, data volume, and I/O tasks.

I. Security in Mobile Cloud

One of the most challenging aspects of CMA is protecting offloaded code and data inside the cloud. While securing contents inside the mobile consumes huge resources, offloading plain contents through insecure wireless medium and storing plain data inside the cloud highly violates user security and privacy. Despite of the large number of research and development in establishing trust in cloud [177]–[180], security and privacy is still one of the major user concerns in utilizing cloud resources that impede CMA deployment. Addressing such crucial needs by employing a novel lightweight security algorithm in mobile side and a set of robust security mechanisms in cloud demand future efforts to promote CMA among smartphone users. In privacy aspects, though recent social behaviors of users in social websites such as Facebook and tweeter advocates that large community of users partially forfeit privacy, they still need certain degree of robust privacy to protect their confidential, clandestine data.

J. Live Virtual Machine Migration

Live migration of VMs between distributed cloud-based servers (especially for distant servers) is a crucial requirement in successfully adopting CMA solutions in MCC, considering wireless network bandwidth and intermittency, and mobility limitations. When a mobile user moves to a place far from the offloaded contents (code or data), the enlarged distance increases access latency and degrades user-observed application performance. Hence, mobilizing the running VM along with the mobile service consumer without perceivable service interruption becomes essential to avoid user experience degradation. However, sharp growing computation and data volume in blipping wireless environment intensifies live migration of VM. Therefore, efforts similar to VMware vMotion [181] and [121], [182] are necessary to optimize VM migration in MCC. Reducing computation complexity and overhead, energy, data volume, and communication cost are critical in low-latency low-cost migration of VM in MCC.

Furthermore, after successful live VM migration, it is essential to ensure that migrated VM is seamlessly accessible via initial IP address when the VM changes its physical machine. Future efforts similar to [183], [184] and LISP [185] are needed to realize the vision of seamless access to migrating VM in MCC.

VIII. Conclusions

Augmenting computing capabilities of mobile devices, especially smartphones using cloud infrastructures and principles is an emerging research area. The ultimate goal of CMA solutions is to realize the vision of unrestricted functionality, storage, and mobility regardless of underlying devices and technologies' constraints. Elasticity, availability, and security

are considered as the most significant characteristics of remote computing resources. Augmenting smartphones significantly improves their usability and adoption in various critical areas such as healthcare, emergency handling, disaster recovery, and crowd management. Enterprise organizations also leverage such value-added services to enhance quality of services delivering to their end-users. However, several communication and networking challenges, particularly live VM migration, seamless back-end and front-end mobility, handover, context awareness, and location management decelerate adoption of cloud-based augmentation solutions in MCC. Similarly, irreconcilable and unpredictable bandwidth, jitter, communication delay, inconsistent network throughput, and non-elastic wireless spectrum and infrastructures encumber upfront planning for CMA.

In this paper, we presented a comprehensive survey on mobile augmentation process and reviewed several CMA approaches. We described deficiencies of current mobile devices as a motivation for augmentation. Comparing surrogates as traditional servers with contemporary cloud-based infrastructures advocates adaptability of cloud computing and principles as an appropriate back-end technology in mobile augmentation domain. Cloud resources in this domain are manifested as distant immobile clouds, proximate mobile and immobile computing entities, and hybrid combination of resources. CMA approaches not only enhance processing, energy, and storage capabilities of mobile devices, but also amend data safety and security, data ubiquity, accessibility, and user interface while facilitate distributed application programming. Despite of significant CMA approaches advantages, several disadvantages hinder their adoption in mobile computing. Several open challenges particularly, autonomic CMA, seamless application mobility, distributed data management, and multipoint data bridging are the most prominent open challenges that demand future efforts.

REFERENCES

[1] M. Weiser, "The computer for the 21st Century.pdf," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 19–25, 2002.

[2] M. Satyanarayanan, "Mobile computing: where's the tofu?" *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 1, no. 1, pp. 17–21, 1997.

[3] S. Abolfazli, Z. Sanaei, and A. Gani, "Mobile Cloud Computing: A Review on Smartphone Augmentation Approaches," in *Proc. WSEAS CISCO'12*, Singapore, 2012.

[4] M. Othman and S. Hailes, "Power conservation strategy for mobile computers using load sharing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 1, pp. 44–51, Jan. 1998.

[5] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 1, pp. 19–26, 1998.

[6] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, 2001.

[7] J. Flinn, D. Narayanan, and M. Satyanarayanan, "Self-tuned remote execution for pervasive computing," in *Proc HotOS'01*, Krün, Germany, 2001, pp. 61–66.

[8] J. Flinn, P. SoYoung, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," in *Proc. IEEE ICDCS '02*, Vienna, Austria, 2002, pp. 217–226.

[9] R. K. Balan, "Simplifying cyber foraging," Doctorate Thesis, Carnegie Mellon University, 2006.

[10] H.-I. Balan, R and Flinn, J and Satyanarayanan, M and S.Sinnamohideen and Yang, "The Case for Cyber Foraging," in *Proc. ACM SIGOPS EW10*, Saint Malo, France, 2002, pp. 87–92.

[11] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb, "Simplifying cyber foraging for mobile devices," in *Proc. ACM MobiSys'07*, Puerto Rico, 2007, pp. 272–285.

[12] R. K. Balan, M. Satyanarayanan, S. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing." in *Proc. ACM MobiSys'03*, San Francisco, California, USA, 2003, pp. 273–286.

[13] S. Goyal and J. Carter, "A lightweight secure cyber foraging infrastructure for resource-constrained devices," in *Proc. MCSA'04*, Lake District National Park, UK, 2004, pp. 186–195.

[14] M. D. Kristensen, "Enabling cyber foraging for mobile devices," in *Proc. MiNEMA'7*, Magdeburg, Germany, 2007, pp. 32–36.

[15] M. D. Kristensen and N. O. Bouvin, "Developing cyber foraging applications for portable devices," in *Proc. 7th IEEE Int'l Conf. Polymers and Adhesives in Microelectronics and Photonics, and 2nd IEEE Int' Interdisciplinary Conf. PORTABLE-POLYTRONIC*, 2008, pp. 1–6.

[16] Y.-Y. Su and J. Flinn, "Slingshot: deploying stateful services in wireless hotspots," in *Proc. ACM MobiSys'05*, Seattle, Washington, USA, 2005, pp. 79–92.

[17] X. Gu and K. Nahrstedt, "Adaptive offloading inference for delivering applications in pervasive computing environments," in *Proc. IEEE PerCom '03*, Dallas-Fort Worth, Texas, USA, 2003.

[18] M. D. Kristensen, "Scavenger: Transparent development of efficient cyber foraging applications," in *Proc. Percom'10*, Mannheim, Germany, 2010, pp. 217–226.

[19] M. Sharifi, S. Kafaie, and O. Kashefi, "A Survey and Taxonomy of Cyber Foraging of Mobile Devices," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–12, 2011.

[20] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[21] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[22] R. Buyya, J. Broberg, and A. Goscinski, *Cloud computing: Principles and paradigms*. Wiley, 2010.

[23] M. Hogan, F. Liu, A. Sokol, and J. Tong, "NIST Cloud Computing Standards Roadmap," Jul. 2011. [Online]. Available: http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Jul5A.pdf

[24] G. Huerta-Canepa and D. Lee, "A Virtual Cloud Computing Provider for Mobile Devices," in *Proc. ACM MCS'10: Social Networks and Beyond*, San Francisco, USA, 2010, pp. 1–5.

[25] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Proc. ACM MobiSys'10*, San Francisco, CA, USA, 2010, pp. 49–62.

[26] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[27] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "AIOLOS: Middleware for improving mobile application performance through cyber foraging," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2629 – 2639, 2012.

[28] S.-H. Hung, C.-S. Shih, J.-P. Shieh, C.-P. Lee, and Y.-H. Huang, "Executing mobile applications on the cloud: Framework and issues," *Computers and Mathematics with Applications*, vol. 63, no. 2, pp. 573–587, 2011.

[29] R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, and H. Bal, "eyeDentify: Multimedia Cyber Foraging from a Smartphone," in *Proc. ISM'09*, San Diego, California, USA, 2009, pp. 392–399.

[30] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," *Proc. IEEE INFOCOM 12*, pp. 945– 953, 2012.

[31] Y. Guo, L. Zhang, J. Kong, J. Sun, T. Feng, and X. Chen, "Jupiter: transparent augmentation of smartphone capabilities through cloud computing," in *Proc. ACM MobiHeld '11*, Cascais, Portugal, 2011, p. 2.

[32] A.Manjunatha, A.Ranabahu, A.Sheth, and K.Thirunarayan, "Power of Clouds In Your Pocket: An Efficient Approach for Cloud Mobile Hybrid Application Development," in *Proc. CloudCom'10*, Indiana, USA, 2010, pp. 496–503.

[33] X. W. 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 & Applications*, vol. 16, no. 3, pp. 270–284, 2011.

[34] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. ACM EuroSys'11*, Salzburg, Austria, 2011, pp. 301–314.

[35] B. G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proc. HotOS'09*, Monte Verità, Switzerland, 2009, pp. 8–14.

[36] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, and B. S. Lee, "μCloud: Towards a New Paradigm of Rich Mobile Applications," in *Proc. IEEE MobWIS'11*, Ontario, Canada, 2011.

[37] X. Luo, "From Augmented Reality to Augmented Computing: a Look at Cloud-Mobile Convergence," *Proc. IEEE ISUVR09*, pp. 29–32, 2009.

[38] E. Badidi and I. Taleb, "Towards a cloud-based framework for context management," in *Proc. IEEE IIT'11*, Abu Dhabi, United Arab Emirates, 2011, pp. 35–40.

[39] Q. Liu, X. Jian, J. Hu, H. Zhao, and S. Zhang, "An Optimized Solution for Mobile Environment Using Mobile Cloud Computing," in *Proc. WiCom'09*, Beijing, China, 2009, pp. 1–5.

[40] K. Kumar and Y. H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[41] B.-G. Chun and P. Maniatis, "Dynamically Partitioning Applications between Weak Devices and Clouds," in *1st ACM Workshop MCS'10: Social Networks and Beyond*, San Francisco, USA, 2010, pp. 1–5.

[42] Y. Lu, S. P. Li, and H. F. Shen, "Virtualized Screen: A Third Element for Cloud-Mobile Convergence," *IEEE Multimedia*, vol. 18, no. 2, pp. 4–11, 2011.

[43] R.Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a Computation Offloading Framework for Smartphones," in *Proc. MobiCASE'10*, Santa Clara, CA, USA, 2010.

[44] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "The Smartphone and the Cloud: Power to the User," in *Proc. MobiCASE '12*, California, USA, 2012, pp. 342–348.

[45] S. Abolfazli, Z. Sanaei, M. Shiraz, and A. Gani, "MOMCC: Market-oriented architecture for Mobile Cloud Computing based on Service Oriented Architecture," in *Proc. IEEE MobiCC'12*, Beijing, China, 2012, pp. 8– 13.

[46] S. Sanaei Z. Abolfazli, M. Shiraz, and A. Gani, "SAMI: Service-Based Arbitrated Multi-Tier Infrastructure Model for Mobile Cloud Computing," in *Proc. IEEE MobiCC'12*, Beijing, China, 2012, pp. 14–19.

[47] R. K. K. Ma and C. L. Wang, "Lightweight Application-Level Task Migration for Mobile Cloud Computing," in *Proc. IEEE AINA'12*, 2012, pp. 550–557.

[48] Y. Gu, V. March, and B. S. Lee, "GMoCA: Green mobile cloud applications," in *Proc. IEEE GREENS'12*, Zurich, Switzerland, Jun. 2012, pp. 15–20.

[49] I. Giurgiu, O. Riva, D. Julic, I. Krivulev, and G. Alonso, "Calling the Cloud: Enabling Mobile Phones as Interfaces to Cloud Applications," *Proc. ACM Middleware09*, vol. 5896, pp. 83–102, 2009.

[50] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Exploiting hardware heterogeneity within the same instance type of Amazon EC2," in *Proc. USENIX HotCloud'12*, Boston, USA, Jun. 2012, p. 4.

[51] Z. Sanaei, S. Abolfazli, A. Gani, and R. K. Buyya, "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges," *IEEE Communications Surveys & Tutorials*, 2013.

[52] K. Salah and R. Boutaba, "Estimating service response time for elastic cloud applications," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 12–16.

[53] J. W. Smith, A. Khajeh-Hosseini, J. S. Ward, and I. Sommerville, "CloudMonitor: Profiling Power Usage," in *Proc. IEEE CLOUD '12*, Jun. 2012, pp. 947–948.

[54] M. Di Francesco, "Energy consumption of remote desktop access on mobile devices: An experimental study," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 105–110.

[55] J. Heide, F. H. P. Fitzek, M. V. Pedersen, and M. Katz, "Green mobile clouds: Network coding and user cooperation for improved energy efficiency," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 111–118.

[56] M. Shiraz, A. Gani, R. Hafeez Khokhar, and R. Buyya, "A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing," *IEEE Communications Surveys & Tutorials*, Nov. 2012.

[57] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 2, pp. 84–106, 2012.

[58] 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*, 2011.

[59] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International journal of high performance computing applications*, vol. 15, no. 3, pp. 200–222, 2001.

[60] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme," in *Proc. ACM CASES '01*, Atlanta, Georgia, USA, 2001, pp. 238–246.

[61] Z. Li and R. Xu, "Energy impact of secure computation on a handheld device," in *Proc. IEEE WWC-5 '02*, Austin, Texas, USA, 2002, pp. 109–117.

[62] A. Athan and D. Duchamp, "Agent-mediated message passing for constrained environments," in *Proc. USENIX MLCS '93*, Cambridge, Massachusetts, 1993, pp. 103–107.

[63] A. Bakre and B. R. Badrinath, "M-RPC: A remote procedure call service for mobile clients," in *Proc. ACM MobiCom'95*, Berkeley, California, 1995, pp. 97–110.

[64] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "The remote processing framework for portable computer power saving," in *Proc. ACM SAC '99*, San Antonio, Texas, USA, 1999, pp. 365–372.

[65] J. M. Wrabetz, D. D. Mason Jr, and M. P. Gooderum, "Integrated remote execution system for a heterogenous computer network environment," Patent US Patent 5,442,791, 1995.

[66] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," *Mobile Networks and Applications*, pp. 1–12, 2012.

[67] K. Bent. (2012, May) Obama: Government Agencies Have One Year To Deploy Smartphone-Friendly Services. [Online]. Available: http://www.crn.com/news/mobility/240000931/obama-government-agencies-have-one-year-to-deploy-smartphone-friendly-services.htm?cid=rssFeed

[68] T. Khalifa, K. Naik, and A. Nayak, "A Survey of Communication Protocols for Automatic Meter Reading Applications," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 168–182, 2011.

[69] M. Satyanarayanan, S. of Computer, Science, C. Mellon, and University, "Mobile Computing: the Next Decade," in *Proc. ACM MCS '10*, San Francisco, USA, 2010.

[70] J. Park and B. Choi, "Automated Memory Leakage Detection in Android Based Systems," *International Journal of Control and Automation*, vol. 5 issue 2, 2012.

[71] G. Xu and A. Rountev, "Precise memory leak detection for java software using container profiling," in *Proc. ACM/IEEE ICSE '08*, Leipzig, Germany, May 2008, pp. 151–160.

[72] M. Satyanarayanan, "Avoiding Dead Batteries," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 2–3, 2005.

[73] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX HotCloud'10*, Boston, MA, 2010, pp. 4–11.

[74] Y. Neuvo, "Cellular phones as embedded systems," *Digest of Technical Papers, IEEE ISSCC '04*, vol. 47, no. 1, pp. 32–37, 2004.

[75] S. Robinson, "Cellphone Energy Gap: Desperately Seeking Solutions," p. 28, 2009. [Online]. Available: http://strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=4645

[76] D. Ben, B. Ma, L. Liu, Z. Xia, W. Zhang, and F. Liu, "Unusual burns with combined injuries caused by mobile phone explosion: watch out for the "mini-bomb!"," *Journal of Burn Care and Research*, vol. 30, no. 6, p. 1048, 2009.

[77] Cellular-News. (2007) Chinese Man Killed by Exploding Mobile Phone. [Online]. Available: http://www.cellular-news.com/story/24754.php

[78] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proc. ACM SOSP' 99*, vol. 33, 1999, pp. 48–63.

[79] T. Starner, D. Kirsch, and S. Assefa, "The Locust Swarm: An environmentally-powered, networkless location and messaging system," in *Proc. IEEE ISWC' 97*, Boston, MA, USA, 1997, pp. 169–170.

[80] S. R.Avro, "Wireless Electricity is Real and Can Change the World," 2009. [Online]. Available: http://www.consumerenergyreport.com/2009/01/15/wireless-electricity-is-real-and-can-change-the-world/

[81] W. F. Pickard and D. Abbott, "Addressing the Intermittency Challenge: Massive Energy Storage in a Sustainable Future [Scanning the Issue]," *Proceedings of the IEEE*, vol. 100, no. 2, pp. 317–321, 2012.

[82] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A Survey of Green Networking Research," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.

[83] N. Vallina-Rodriguez and J. Crowcroft, "Energy Management Techniques in Modern Mobile Handsets," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 179–198, 2013.

[84] K. Jackson. (2009) Missouri University Researchers Create Smaller and More Efficient Nuclear Battery. [Online]. Available: http://munews.missouri.edu/news-releases/2009/1007-mu-researchers-create-smaller-and-more-efficient-nuclear-battery/

[85] J. F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva, "The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011," White Paper, 2008.

[86] X. F. Guo, J. J. Shen, and S. M. Wu, "On Workflow Engine Based on Service-Oriented Architecture," *Proc. IEEE ISISE '08*, pp. 129–132, 2008.

[87] S. Ortiz, "Bringing 3D to the Small Screen," *IEEE Computer*, vol. 44, no. 10, pp. 11–13, 2011.

[88] T. Capin, K. Pulli, and T. Akenine-Moller, "The State of the Art in Mobile Graphics Research," *IEEE Computer Graphics and Applications*, vol. 28, no. 4, pp. 74–84, 2008.

[89] Prosper Mobile Insights, "Smartphone/Tablet User Survey," 2011. [Online]. Available: http://prospermobileinsights.com/Default.aspx?pg=19

[90] M. La Polla, F. Martinelli, and D. Sgandurra, "A Survey on Security for Mobile Devices," *IEEE Communications Surveys & Tutorials*, 2012.

[91] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.

[92] C. Cachin and M. Schunter, "A cloud you can trust," *IEEE Spectrum*, vol. 48, no. 12, pp. 28–51, Dec. 2011.

[93] NVidia. (2010) The Benefits of Multiple CPU Cores in Mobile Devices. [Online]. Available: http://www.nvidia.com/content/PDF/tegra_white_papers/Benefits-of-Multi-core-CPUs-in-Mobile-Devices_Ver1.2.pdf

[94] ARM. (2009) The ARM Cortex-A9 Processors Version 2.0. White Paper. [Online]. Available: http://arm.com/files/pdf/ARMCortexA-9Processors.pdf

[95] M. Altamimi, R. Palit, K. Naik, and A. Nayak, "Energy-as-a-Service (EaaS): On the Efficacy of Multimedia Cloud Computing to Save Smartphone Energy," in *Proc. IEEE CLOUD '12*, Honolulu, Hawaii, USA, Jun. 2012, pp. 764–771.

[96] K. Fekete, K. Csorba, B. Forstner, M. Feher, and T. Vajk, "Energy-efficient computation offloading model for mobile phone environment," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 95–99.

[97] S. Park and B.-S. Moon, "Design and Implementation of iSCSI-based Remote Storage System for Mobile Appliance," *Proc. IEEE HEALTHCOM '05*, pp. 236–240, 2003.

[98] G. Lawton, "Cloud Streaming Brings Video to Mobile Devices," *IEEE Computer*, vol. 45, no. 2, pp. 14–16, 2012.

[99] B. Seshasayee, R. Nathuji, and K. Schwan, "Energy-aware mobile service overlays: Cooperative dynamic power management in distributed mobile systems," in *Proc. IEEE ICAC'07*, Jacksonville, Florida, USA, 2007, p. 6.

[100] Y. J. Hong, K. Kumar, and Y. H. Lu, "Energy efficient content-based image retrieval for mobile systems," in *Proc. IEEE ISCAS '09*, Taipei, Taiwan, 2009, pp. 1673–1676.

[101] B. Rajesh Krishna, "Powerful change part 2: reducing the power demands of mobile devices," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 71–73, 2004.

[102] A. F. Murarasu and T. Magedanz, "Mobile middleware solution for automatic reconfiguration of applications," in *Proc. ITNG '09*, Las Vegas, Nevada, USA, 2009, pp. 1049–1055.

[103] E. Abebe and C. Ryan, "Adaptive application offloading using distributed abstract class graphs in mobile environments," *Journal of Systems and Software*, vol. 85, no. 12, pp. 2755–2769, 2012.

[104] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 4, no. 2, p. 14, 2009.

[105] G. Huerta-Canepa and D. Lee, "An adaptable application offloading scheme based on application behavior," in *Proc. IEEE AINAW '08*, GinoWan, Okinawa, Japan, 2008, pp. 387–392.

[106] F. Schmidt, *The SCSI bus and IDE interface: protocols, applications and programming*. Addison-Wesley Longman Publishing Co., Inc., 1997.

[107] Y. Lu and D. H. C. Du, "Performance study of iSCSI-based storage subsystems," *IEEE Communications Magazine*, vol. 41, no. 8, pp. 76–82, 2003.

[108] J.-H. Kim, B.-S. Moon, and M.-S. Park, "MiSC: A New Availability Remote Storage System for Mobile Appliance," in *ICN 2005*, ser. LNCS 3421, P. Lorenz and P. Dini, Eds. Springer, 2005, pp. 504 – 520.

[109] M. Ok, D. Kim, and M. Park, "UbiqStor: Server and Proxy for Remote Storage of Mobile Devices," *Emerging Directions in Embedded and Ubiquitous Computing*, pp. 22–31, 2006.

[110] M. H. OK, D. Kim, and M. Park, "UbiqStor: A remote storage service for mobile devices," *Parallel and Distributed Computing: Applications and Technologies*, pp. 53–74, 2005.

[111] D. Kim, M. Ok, and M. Park, "An Intermediate Target for Quick-Relay of Remote Storage to Mobile Devices," in *Computational Science and Its Applications - ICCSA 2005*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3481, pp. 91–100.

[112] W. Zheng, P. Xu, X. Huang, and N. Wu, "Design a cloud storage platform for pervasive computing environments," *Cluster Computing*, vol. 13, no. 2, pp. 141–151, 2010.

[113] U. Kremer, J. Hicks, and J. Rehg, "A compilation framework for power and energy management on mobile computers," *Languages and Compilers for Parallel Computing*, pp. 115–131, 2003.

[114] S. Gurun and C. Krintz, "Addressing the energy crisis in mobile computing with developing power aware software," vol. 8, no. 64MB, 2003. [Online]. Available: http://www.cs.ucsb.edu/research/tech_reports/reports/2003-15.ps

[115] X. Ma, Y. Cui, and I. Stojmenovic, "Energy Efficiency on Location Based Applications in Mobile Cloud Computing: A Survey," *Procedia Computer Science*, vol. 10, pp. 577–584, 2012.

[116] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," in *Proc. IEEE VTC Spring '11*, Budapest, Hungary, 2011, pp. 1–6.

[117] E. D. Lara, D. S. Wallach, and W. Zwaenepoel, "Puppeteer: Component-based adaptation for mobile computing," in *Proc. USENIX USITS*, 2001, pp. 14–25.

[118] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in *Proc. ACM OOPSLA' 09*, Orlando, Florida, USA, 2009, pp. 627–634.

[119] M. Shiraz, S. Abolfazli, Z. Sanaei, and A. Gani, "A study on virtual machine deployment for application outsourcing in mobile cloud computing," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 946–964, Dec. 2012.

[120] Z. Sanaei, S. Abolfazli, A. Gani, and R. H. Khokhar, "Tripod of Requirements in Horizontal Heterogeneous Mobile Cloud Computing," in *Proc. WSEAS CISCO'12*, 2012.

[121] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster Computing*, pp. 1–16, 2011.

[122] B. Gao, L. He, L. Liu, K. Li, and S. A. Jarvis, "From Mobiles to Clouds: Developing Energy-Aware Offloading Strategies for Workflows," in *Proc. ACM/IEEE GRID '12*, Dubna, 2012, pp. 139–146.

[123] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in *Proc. ACM ICIS '09*, Seoul, Republic of Korea, 2009, pp. 1044–1048.

[124] J. Yang, H. Wang, J. Wang, C. Tan, and D. Yu, "Provable Data Possession of Resource-constrained Mobile Devices in Cloud Computing," *Journal of Networks*, vol. 6, no. 7, pp. 1033–1040, 2011.

[125] N. Nasser, A. Hasswa, and H. Hassanein, "Handoffs in fourth generation heterogeneous networks," *IEEE Communications Magazine*, vol. 44, no. 10, pp. 96–103, 2006.

[126] "Mobile Cloud - Converged Fix, Wireless and Mobile Infrastructure," 2013. [Online]. Available: http://sieneon.com/infras.htm

[127] "Converged Highly Integrated Network Access (CHINA)," 2013. [Online]. Available: http://sieneon.com/network.htm

[128] R. Kuhne, G. Huitema, and G. Carle, "Charging and Billing in Modern Communications Networks A Comprehensive Survey of the State of the Art and Future Requirements," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 170–192, 2012.

[129] (2013) Panda Global Business Protection on-premise security solutions. [Online]. Available: http://www.pandasecurity.com/usa/enterprise/security-solutions-on-premise/

[130] S. Kamara and K. Lauter, "Cryptographic cloud storage," *Financial Cryptography and Data Security*, pp. 136–149, 2010.

[131] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *Proc. IEEE IWQoS '09*, Charleston, South Carolina, USA, 2009, pp. 1–9.

[132] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. O'Reilly Media, Incorporated, 2009.

[133] J. Clark. (2013, Feb.) Microsoft secure Azure Storage goes down Wordwide. [Online]. Available: http://www.theregister.co.uk/2013/02/22/azure_problem_that_should_never_happen_ever/

[134] A. Clark, Christopher and Fraser, Keir and Hand, Steven and Hansen, Jacob Gorm and Jul, Eric and Limpach, Christian and Pratt, Ian and Warfield, "Live migration of virtual machines," in *Proc. USENIX NSDI' 05*, Boston, MA, USA, 2005, pp. 273–286.

[135] K. Owens. Securing Virtual Compute Infrastructure in the Cloud. Whitepaper. [Online]. Available: http://www.savvis.com/en-us/info_center/documents/hos-whitepaper-securingvirutalcomputeinfrastructureinthecloud.pdf

[136] S. Garriss, R. Cáceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang, "Trustworthy and personalized computing on public kiosks," in *Proc. ACM MobiSys '08*, Breckenridge, CO, USA, 2008, pp. 199–210.

[137] N. D. Lane, E. Miluzzo, L. Hong, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[138] P. Lukowicz, S. Pentland, and A. Ferscha, "From Context Awareness to Socially Aware Computing," *Pervasive Computing*, vol. 11, no. 1, pp. 32–41, Jan. 2012.

[139] P. Makris, D. N. Skoutas, and C. Skianis, "A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, 2013.

[140] S. Shen and B. Blau. (2012, Aug.) Market Trends: Mobile App Stores, Worldwide, 2012. [Online]. Available: http://www.gartner.com/id=2126015

[141] M. ANIA. (2012) Beware: Free Apps Might Prove Costly. [Online]. Available: http://theinstitute.ieee.org/technology-focus/technology-topic/beware-free-apps-might-prove-costly

[142] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *Proc. IEEE CCS'11*, Milan, Italy, 2011, pp. 235–245.

[143] M. Rahimi, N. Venkatasubramanian, S. Mehrotra, and V. Vasilakos, "MAPCloud: mobile applications on an elastic and scalable 2-tier cloud architecture," in *Proc. IEEE/ACM UCC'12*, Chicago, Illionis , USA, 2012.

[144] H. Kim and M. Parashar, "CometCloud: An Autonomic Cloud Engine," *Cloud Computing: Principles and Paradigms*, pp. 275–297, 2011.

[145] J. Broberg, R. Buyya, and Z. Tari, "MetaCDN: Harnessing Storage Clouds for high performance content delivery," *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1012–1022, 2009.

[146] K. Hariharan, "Best Practices: Extending Enterprise Applications to Mobile Devices," *The Architecture Journal, Microsoft Architecture Center*, vol. 14, 2008. [Online]. Available: http://msdn.microsoft.com/en-us/library/bb985493.aspx

[147] A. H. Ranabahu, E. M. Maximilien, A. P. Sheth, and K. Thirunarayan, "A domain specific language for enterprise grade cloud-mobile hybrid applications," in *Proc. ACM co-located workshops on DSM'11, TMC'11, AGERE'11, AOOPES'11, NEAT'11, & VMIL'11*, Portland, Oregon, USA, 2011, pp. 77–84.

[148] A. van Deursen, P. Klint, and J. Visser, "Domain-Specific Languages: An Annotated Bibliography," *SIGPLAN Notices*, vol. 35, no. 6, pp. 26–36, 2000.

[149] P. Stuedi, I. Mohomed, and D. Terry, "WhereStore: Location-based data storage for mobile devices interacting with the cloud," in *Proc. ACM MCS'10*, San Francisco, California, USA, 2010, p. 1.

[150] H. Mao, N. Xiao, W. Shi, and Y. Lu, "Wukong: A cloud-oriented file service for mobile Internet devices," *Journal of Parallel and Distributed Computing*, 2011.

[151] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," *IEEE Computer*, vol. 39, no. 3, pp. 46–52, 2006.

[152] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce," Master Thesis, Computer Science Deptartment, Carnegie Mellon University, 2009.

[153] C. Mei, D. Taylor, C. Wang, A. Chandra, and J. Weissman, "Mobilizing the Cloud: Enabling Multi-User Mobile Outsourcing in the Cloud," Department of Computer Science and Engineering, University of Minnesota, Tech. Rep. TR 11-029, 2011.

[154] C. Mei, D. Taylor, C. Wang, A. Chandra, and J. Weissman, "Sharing-Aware Cloud-Based Mobile Outsourcing," in *Proc. IEEE CLOUD '12*, Hawaii, USA, Jun. 2012, pp. 408–415.

[155] M. Guirguis, R. Ogden, Z. Song, S. Thapa, and Q. Gu, "Can You Help Me Run These Code Segments on Your Mobile Device?" in *Proc. IEEE GLOBECOM '11*, Houston, Texas, USA, Dec. 2011, pp. 1–5.

[156] T. White, *Hadoop: The definitive guide*. O'Reilly Media, 2012.

[157] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2004.

[158] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE ISCC '12*, Cappadocia, Turkey, Jul. 2012, pp. 59–66.

[159] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proc. ACM MCS '12*, ser. MCS '12, Helsinki, Finland, 2012, pp. 21–28.

[160] Gartner. (2012, Feb.) Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1924314

[161] H. Viswanathan, B. Chen, and D. Pompili, "Research Challenges in Computation, Communication, and Context awareness for Ubiquitous Healthcare," *IEEE Communications Magazine*, vol. 50, no. 5, p. 92, 2012.

[162] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging Smartphone Cameras for Collaborative Road Advisories," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 707–723, May 2012.

[163] M. Kranz, A. Möller, N. Hammerla, S. Diewald, T. Plötz, P. Olivier, and L. Roalter, "The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices," *Pervasive and Mobile Computing*, vol. 9, no. 2, pp. 203–215, 2012.

[164] P. Bellavista, A. Corradi, and C. Giannelli, "A unifying perspective on context-aware evaluation and management of heterogeneous wireless connectivity," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 337–357, 2011.

[165] American National Standard Dictionary of Information Technology. [Online]. Available: http://incits.org/ANSDIT/p2.htm[June-19-2012]

[166] X. Yan, Y. Ahmet Sekercioglu, and S. Narayanan, "A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks," *Computer Networks*, vol. 54, no. 11, pp. 1848–1863, 2010.

[167] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 311–336, 2011.

[168] H. Lagar-Cavilla, N. Tolia, E. De Lara, M. Satyanarayanan, and D. O'Hallaron, "Interactive resource-intensive applications made easy," in *Proc. MIDDLEWARE*. Springer, 2007, pp. 143–163.

[169] R. Bifulco and R. Canonico, "Analysis of the handover procedure in Follow-Me Cloud," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 185–187.

[170] D. Johansson and K. Andersson, "Web-based adaptive application mobility," in *Proc. IEEE CLOUDNET'12*, Paris, Nov. 2012, pp. 87–94.

[171] B. C. Guevara, Marisabel and Lubin, Benjamin and Lee, "Navigating Heterogeneous Processors with Market Mechanisms," in *Proc. IEEE HPCA'13*, Shenzhen, China, 2013.

[172] R. Buyya, R. N. Calheiros, and X. Li, "Special Section on Autonomic Cloud Computing: Technologies, Services, and Applications," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 9, pp. 935–937, 2012.

[173] P. Yu, X. Ma, J. Cao, and J. Lu, "Application mobility in pervasive computing: A survey," *Pervasive and Mobile Computing*, vol. 9, no. 1, pp. 2–17, Feb. 2013.

[174] G. Blair, M. Paolucci, P. Grace, and N. Georgantas, "Interoperability in Complex Distributed Systems," in *Formal Methods for Eternal Networked Software Systems*, ser. Lecture Notes in Computer Science, M. Bernardo and V. A. Issarny, Eds. Springer Berlin Heidelberg, 2011, vol. 6659, pp. 1–26.

[175] A.Ranabahu and A.Sheth, "Semantics Centric Solutions for Application and Data Portability in Cloud Computing," in *Proc. IEEE CloudCom'10*, Indianapolis, USA, 2010, pp. 234–241.

[176] L. Youseff, R. Wolski, B. Gorda, and C. Krintz, "Paravirtualization for HPC Systems," in *Frontiers of High Performance Computing and Networking-ISPA 2006 Workshops*, ser. Lecture Notes in Computer Science, G. Min, B. Martino, L. Yang, M. Guo, and G. Ruenger, Eds., 2006, vol. 4331, pp. 474–486.

[177] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," *Proc. Springer ESORICS '09*, pp. 355–370, 2009.

[178] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in *Proc. ACM CCS' 09*, Chicago, IL, USA, 2009, pp. 55–66.
[179] M. Mowbray and S. Pearson, "A client-based privacy manager for cloud computing," in *Proc. ACM COMSWARE '09*, Dublin, Ireland, 2009, p. 5.
[180] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *Proc. IEEE TRUSTCOM ' 11*, Changsha, China, 2011, pp. 91–98.
[181] "Virtual Machine Migration Comparision: VMwareVsphere VS. Microsoft Hyper-V," p. 36, 2011. [Online]. Available: http://www.vmware.com/files/pdf/vmw-vmotion-verus-live-migration.pdf
[182] T. Takahashi, Kazushi and Sasada, Koichi and Hirofuchi, "A Fast Virtual Machine Storage Migration Technique Using Data Deduplication," in *Proc. CLOUD COMPUTING'12*, Nice, France, 2012, pp. 57–64.
[183] R. Watanabe, Hidenobu and Ohigashi, Toshihiro and Kondo, Tohru and Nishimura, Kouji and Aibara, "A Performance Improvement Method for the Global Live Migration of Virtual Machine with IP Mobility," in *Proc. IEEE/IPSJ ICMU '10*, Seattle, Washington, USA, 2010.
[184] M. Harney, Eric and Goasguen, Sebastien and Martin, Jim and Murphy, Mike and Westall, "The efficacy of live virtual machine migrations over the internet," in *Proc. ACM VTDC '07*, Reno, Nevada, USA, 2007.
[185] G. Raad, Patrick and Colombo, Giulio and Chi, Dung Phung and Secci, Stefano and Cianfrani, Antonio and Gallard, Pascal and Pujolle, "Achieving Sub-Second Downtimes in Internet-wide Virtual Machine Live Migrations in LISP Networks," in *Proc. IFIP/IEEE Symposium on Integrated Network Management*, Ghent, Belgium, 2013.

Saeid Abolfazli is currently a Ph.D. candidate, research assistant in High Impact Research Project (Mobile Cloud Computing: Device and Connectivity) fully funded by Malaysian Ministry of Higher Education, and part time lecturer in the Department of Computer Systems and Technology at the University of Malaya, Malaysia. He received his M.Sc in Information Systems in 2008 from India and BE (Software Engineering) in 2001 from Iran. He has been serving as CEO of Espanta Information Complex during 1999-2006 in Iran. He also was part time lecturer to the ministry of education and Khorasan Technical and Vocational Organization between 2000 and 2006. He is a member of IEEE society and IEEE CS Cloud Computing STC. He has been serving as a reviewer for several international conference and ISI journals of computer science. His main research interests include Mobile Cloud Computing, lightweight protocols, and service oriented computing (SOC). Please write to him at abolfazli.s@gmail.com or abolfazli@ieee.org. For further information, please visit his cyberhome at www.mobilecloudfamily.com

Zohreh Sanaei is currently a Ph.D. candidate and research assistant in High Impact Research Project (Mobile Cloud Computing: Device and Connectivity) fully funded by Malaysian Ministry of Higher Education in the Department of Computer Systems and Technology at the University of Malaya, Malaysia. She received her M.Sc. in Information Systems in 2008 from India and BE (Software Engineering) in 2001 from Iran. She worked in 3MCD and EIC, Iran as a network engineer and participated in several wireless communication projects from 2001 till 2006. She has been working for more than 6 years as a part-time lecturer for the ministry of social affairs, Iran as a technical and vocational trainer. Her main research interests include mobile cloud computing, distributed computing, and ubiquitous computing. She is a member of IEEE society and can be corresponded via zsanaeim@gmail.com or sanaei@ieee.org. For further information, please visit her cyberhome at www.mobilecloudfamily.com

Ejaz Ahmed was born in Gandhian, Mansehra, Pakistan. He did his B.S (Computer Science) from Allama Iqbal Open University, Islamabad, Pakistan. Afterward, he completed his M.S (Computer Science) from Mohammad Ali Jinnah University, Islamabad in 2009. Currently, he is pursuing his PhD Candidature under Bright Spark Program at Faculty of Computer and Information Technology, University Malaya, Kuala Lumpur, Malaysia. He is an active researcher in Mobile Cloud Computing Research Group at Faculty of Computer Science and Information Technology, University Malaya, Kuala Lumpur, Malaysia. His areas of interest include Seamless Application Execution Framework Design for Mobile Cloud Computing, Designing of Channel Assignment and Routing Algorithms for Cognitive Radio Networks.

Dr. Abdullah Gani is Associate Professor of Computer System and Technology at the University of Malaya, Malaysia. His academic qualifications were obtained from UK's universities - bachelor and master degrees from the University of Hull, and Ph.D from the University of Sheffield. He has vast teaching experience due to having worked in various educational institutions locally and abroad - schools, teaching college, ministry of education, and universities. His interest in research started in 1983 when he was chosen to attend Scientific Research course in RECSAM by the Ministry of Education, Malaysia. More than 100 academic papers have been published in conferences and respectable journals. He actively supervises many students at all level of study - Bachelor, Master and PhD. His interest of research includes self-organized systems, reinforcement learning, and wireless-related networks. He is now working on mobile cloud computing with High Impact Research Grant for the period of 2011-2016.

Dr. Rajkumar Buyya is Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council, and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 425 publications and four text books including "Mastering Cloud Computing" published by McGraw Hill and Elsevier/Morgan Kaufmann, 2013 for Indian and international markets respectively. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index=70, g-index=144, 23000+ citations). Microsoft Academic Search Index ranked Dr. Buyya as the world's top author in distributed and parallel computing between 2007 and 2012. Recently, ISI has identified him as a "Highly Cited Researcher" based on citations to his journal papers.

Software technologies for Grid and Cloud computing developed under Dr. Buyya's leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE/ACM conferences. These contributions and international research leadership of Dr. Buyya are recognized through the award of "2009 IEEE Medal for Excellence in Scalable Computing" from the IEEE Computer Society, USA. Manjrasoft's Aneka Cloud technology developed under his leadership has received "2010 Asia Pacific Frost & Sullivan New Product Innovation Award" and "2011 Telstra Innovation Challenge, People's Choice Award". He is currently serving as the foundation Editor-in-Chief (EiC) of IEEE Transactions on Cloud Computing. For further information on Dr. Buyya, please visit his cyberhome: www.buyya.com