Received June 1, 2016, accepted June 8, 2016, date of publication September 21, 2016, date of current version January 23, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2602321

Mobilouds: An Energy Efficient MCC Collaborative Framework With Extended Mobile Participation for Next Generation Networks

JOHN PANNEERSELVAM[1], (Member, IEEE), JAMES HARDY[1], LU LIU[1], (Member, IEEE), BO YUAN[1,2], AND NICK ANTONOPOULOS[1]

[1]Department of Engineering and Technology, University of Derby, Derby, DE22 1GB, U.K.

[2]Key Laboratory of Embedded Systems and Service Computing, Ministry of Education, Tongji University, Shanghai 100044, China

Corresponding author: L. Liu (l.liu@derby.ac.uk)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502209 and Grant 61502207, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20130528.

ABSTRACT Given the emergence of mobile cloud computing (MCC), its associated energy implications are witnessed at larger scale. With offloading computationally intensive tasks to the cloud datacentres being the basic concept behind MCC, most of the mobile terminal resources participating in the MCC collaborative execution are wasted as they remain idle until the mobile terminals receive responses from the datacentres. This is an additional wastage of resources alongside the cloud resources are already being addressed as massive energy consumers. Though the energy consumed of the idle mobile resources is insignificant in comparison with the cloud counterpart, such consumptions have drastic impacts on the mobile devices causing unnecessary battery drains. To this end, this paper proposes Mobilouds which encompass a multi-tier processing architecture with various levels of process cluster capacities and a software application to manage energy efficient utilization of such process clusters. Our proposed Mobilouds framework encourages the mobile device participation in the MCC collaborative execution, thereby reduces the presence of idle mobile resources and utilizes such idle resources in the actual task execution. Our performance evaluation results demonstrate that the Mobilouds framework offers the most energy-time balancing process clusters for task execution by effectively utilizing the available resources, in comparison with an entire cloud offloading strategy using 5G/4G networks.

INDEX TERMS Cloudlets, consumption, idleness, offloading.

I. INTRODUCTION

Mobile Cloud Computing (MCC) is an integrated framework encompassing cloud datacentres, mobile devices and communication infrastructure. MCC deployments are widely being witnessed in various applications [1], [2] such as telemonitoring, e-learning, tele-surgery, IT, business services, rural and urban development etc. MCC service models [3] involve the establishments of complex relationships between infrastructure providers, application and service providers, developers and end-users. Infrastructure providers generally provide supplements for hardware and software services; application providers are responsible for executing user requested services; developers are generally the consumers of cloud services who develop applications being hosted on the cloud datacentres; and end-users are the consumers of the MCC services. The end-users of the MCC services generally do not have the privilege of control over the underlying infrastructures such as hardware, network, servers etc., but they can have a complete privilege over the applications deployed by them. The success of such an MCC framework depends on the seamless integration of the dedicated hardware and software resources of the three core technologies. But the heterogeneities [4] found across these three cross platforms impose several practical challenges in developing an energy efficient integrated MCC infrastructure.

The concept of MCC is developed with the vision of transforming the compute capacities [5], [6] of resource poor-mobile devices into resource-rich computing components, with the cloud datacentres facilitating the necessary process supplements. The limitations of mobile devices [7] can be attributed to their limited battery life, intrinsic process limitations such as compute cores, storage etc., and extrinsic limitations including transmitting and receiving capabilities etc. MCC service concept can be viewed from two different perspectives [8] such as the simple viewpoint and the mobile device viewpoint. The former involves data processing and storage only in the cloud datacentre outside of the mobile devices, and the later involves data processing and storage held in the mobile devices supported by the cloud datacentres. Similarly, MCC implementation [9] can be viewed from two different perspectives. One is the infrastructure based implementation, where cloud services are provided to the mobile users over a static hardware infrastructure. Second is the ad-hoc mobile cloud based implementation, in which a group of mobile devices forms a process cluster resembling a miniature cloud for the purpose of sharing the process supplements among the participating mobile devices.

The concept behind most of the mobile cloud collaborative execution frameworks is to offload the computationally intensive tasks from the resource poor mobile terminals to the resource rich cloud datacentres. Offloading all the tasks from the mobile terminals to the cloud datacentres is the strategy of a typical Cloud Computing service model, which involves processing the workloads entirely in the cloud datacenters [10]–[12]. Usually such a kind of complete offloading is achieved when the computational requirements exceed the process capacities of the mobile terminals. An efficient MCC service model should incorporate the participation of the mobile terminals rather than entirely relying on the datacentres. Communication infrastructure which bridges the mobile devices with the cloud backend servers plays crucial roles in the overall performance of the MCC infrastructure, thus significantly impacts the overall energy efficiency.

Existing MCC offloading frameworks promoting energy efficiency include techniques such as avalanche offloading, middleware based task scheduling, opportunistic offloading, trans-receiver switching, and strategies of conserving transmission energy etc. Though all such works are promoting energy efficiency, they are witnessed to be conserving the mobile resources rather than utilising such mobile terminals in the mobile cloud collaborative execution. Further, the aforementioned approaches are leaning towards the cloud efficiency rather than developing an integrated energy efficient solution. In general undesirable energy consumptions in an MCC infrastructure are evident across all the three integrated core technologies. In most of these approaches, mobile terminals generally should wait until the tasks are offloaded, processed, and responded back from the cloud datacentres. This strategy leaves the mobile resources idle [13] without any process contribution towards task execution, and such resources are wasted until the task is executed in the cloud datacentres. The waiting time of the users in an accumulation of the offloading time, processing time and the response time consumed by the communication and the cloud providers. This waiting time is a dominating factor in determining the Quality of Service (QoS) and in satisfying the Quality of Expectation (QoE) of the users. Energy consumption and service delays are directly related in such a way that to reduce the service delays providers usually tend to consume excess resources, resulting in excess energy consumptions.

Rather than conserving the mobile resources and leaving them idle, extending the participation of the mobile terminals in the MCC collaborative execution can provide mobile backend services for the cloud servers in accordance with the capacities of the mobile devices. Encouraging the participation of the mobile terminals facilitates MCC to deliver user-centric compute capabilities for task executions depending on real-time availability. Augmented mobile resources [5] provided with cloud supplements virtually increases the compute capacities of the resource poor mobile devices. Thus the MCC collaborative execution strategy with extended mobile participation can effectively reduce the presence of idle resources at the mobile terminals. To this end, this paper addresses the issues of energy wastages incurred by the idle mobile resources and proposes a hybrid energy efficient MCC architecture, named Mobilouds, for the purpose of extending the participation of the mobile devices in the collaborative MCC execution ultimately to reduce the undesirable energy consumptions of the idle mobile resources with reduced service delays. The major contributions of this paper include,

1. A novel multi-tier process architecture named Mobilouds, composed of various capacities of process clusters for energy efficient MCC collaborative execution. This Mobilouds architecture can be both upgraded to a higher capacity cluster during resource scarcity and downgraded when there is excess resources in the process cluster for the purpose of conserving energy.

2. The Mobilouds application which is a software process deployed to facilitate the functionalities of the Mobilouds architecture. This software process runs in the mobile devices for computing the resource availability in the mobile terminals. This Mobilouds process helps to select the optimum process cluster from the Mobilouds architecture, and an energy efficient MCC collaborative execution is achieved in the chosen process cluster by the way of a deploying a distributed offloading strategy among the available resources in the cluster.

The remainder of the paper is organised as follows. Section II presents the related works and Section III gives a background study on factors affecting energy efficiency in MCC offloading. Section IV describes our proposed Mobilouds framework and Section V illustrates the optimum cluster selection process of Mobilouds. The implementation of our proposed Mobilouds is presented in Section VI. Section VII includes the performance evaluations of our Mobilouds framework based on a real file scenario. Section VIII presents the applicability of our proposed Mobilouds. Our future research plans are presented in section IX and section X concludes this paper.

II. RELATED WORKS

MCC is growing importance in the recent past and a notable number of MCC offloading frameworks have been proposed for energy efficient MCC processing. Task Offloading using

Self Organized Criticality (TOSOC) [14] is an offloading framework based on Avalanche offloading, where tasks are executed in the process clusters formed of mobile devices. New mobile terminals are invited when the current cluster overloads, thus tasks are offloaded to a chain of mobile terminals until either the task is completed or the current cluster overloads. In spite of the computational variations found among the mobile devices, deciding the threshold for inviting new terminals is always challenging. Also the availability of new mobile terminals are uncertain, thus guaranteed execution of the tasks is not often achieved. TOSOC does not allow a given mobile terminal to offload tasks more than once to avoid the ping-pong effect. This prevents the framework from being more scalable and flexible during process failures and data losses. TOSOC increases the energy consumptions whilst reducing service delays. Avalanche approach may not scale well in a dynamic MCC environment, since tasks are bound to different computational complexities and mobile terminals are static with their processing capacities, and so elasticity in the process clusters can hardly be achieved.

A task scheduler model based on a centralised broker [15] has been developed to optimally offload tasks to the datacentres for optimising energy consumption. With the mobile devices connected to the centralised broker, the broker manages the task offloading based on a defined threshold for every task execution. Being a centralised controlled system, the failure of the centralised broker will lead to the failure of the entire system. Also, this model is aimed at reducing the energy consumption incurred only across the participating mobile devices.

The issue of sub-optimal offloading [16] resulting from network inconsistency is resolved with a three-tier architecture by predicting the real time resource availability. This approach is enabled with a Wi-Fi access point selection scheme to find the most energy efficient solution. Though this approach considers user mobility and server workload balance management, the applicability of such methods is not scalable in remote or isolated regions under cellular networks.

The approach of improving the energy efficiency of the overlay transmission time during offloading [7] has been proposed by exploiting parallelism and VM synthesis using a higher bandwidth short range wireless network to reduce handoff delays. Though this approach offers high speed processing and energy reduction, parallelism introduces issues in data synchronisation and data sharing. Furthermore, the inter-dependencies among the tasks do not always allow full parallelism. Also, in mobile devices the endpoint terminals are not often optimised for higher performance, thus not scalable for higher transmission capabilities.

An opportunistic offloading [17], [18] strategy has been developed with the motivation of enabling workload migration from mobile terminals located in remote regions to the cloud datacentres. This strategy triggers workload migration whenever the nearby mobile terminals come in contact. Though such strategies help offloading in isolated regions, both time and energy efficiency are not often achieved. Furthermore, the encounters among the mobile devices are not always guaranteed during resource requests.

Few other existing energy efficient approaches in mobile-cloud communication [7] include turning on and off the trans-receiver, and exploiting low power [19] states. But such approaches affects the bandwidth utility, and the wake-up latencies incurred during the state transition are often non-negligible. Since the reception of the cloud response is not previously known, this approach requires the trans-receiver to be turned on until the job response is received, thereby incurring undesirable trans-receiver energy consumption.

Phone-to-cloud [20] is an offloading strategy proposed to conserve the mobile energy enabled with an offloading decision model. This decision model decides upon the offloading based on predicting the execution duration, CPU efficiency and the bandwidth availability. Cloud datacentre resources not necessarily be static and obtaining contextual clues might help effective decision making rather than prediction. A code/task offloading strategy [21] has been proposed to reduce the energy costs of the workflows between mobile devices and the cloud by coordinating the mobile devices to the resources in the cloudlets.

A scheduling policy [22] has been devised for collaborative execution between mobile devices and the cloud, by the way of formulating the minimum-energy task scheduling problem as a constrained short path problem on a directed acyclic graph to conserve the mobile energy. Etime [23] is a data transmission strategy between mobile devices and cloud based on Lyapunov optimisation, proposed to combat intermittent networks for energy efficiency. A semi-Markov decision process (SMDP)-based optimisation framework [24] has been proposed based on Dynamic Voltage Frequency Scaling (DVFS) to conserve energy drawn from mobile battery drains incurred during service requests. All such works are witnessed to be conserving the mobile resources by offloading the computation to the datacentres. Such approaches are vulnerable to leave the mobile resource idle, effective utilisation of such resources helps promoting energy efficiency across the three integrated core technologies.

Form the state-of-the art offloading techniques, MCC still demands a complete energy efficient collaborative execution framework. Most of the existing approaches are leaving the energy consumptions incurred by the idle resources of the native mobile devices unnoticed. Though such energy levels are insignificant when compared to the cloud counter-part, utilising such idle resources is important for the mobile standards which would not only increase the battery efficiency of the mobile devices but also reduces the overall execution time and energy consumptions of the MCC infrastructure. This necessitates the demand for extending the participation of the mobile terminals by the way of developing a collaborative execution framework between the mobile terminals and the datacentres.

III. BACKGROUND

This sections elaborates on the underpinning conceptions in MCC offloading along with the concerns and challenges imposed by the underlying components on energy efficiency.

Communicating or Offloading tasks to the distant cloud datacentres from the mobile devices is governed by several internal and external factors [25], [26] in the MCC infrastructure. The efficiency of the MCC infrastructure directly depends on the characteristics of the wireless network [27], [28]. Most often, propagation and transmission delays in the network are relatively longer [13] and fluctuant incurring adverse energy consumptions. Despite the deployments of 4G being witnessed in several countries, the increasing demand for energy conservation whilst achieving high data and mobility rates have shifted the interest of MCC providers towards 5G [29] in the recent past. Communication delays usually incur additional and undesirable wait times for the users, thereby considerably affecting the QoS. Standards such as LTE/LTE-advanced [3] are now being deployed with MCC-based adaptive regulation mechanisms to achieve effective QoS and QoE.

Distance of the mobile terminals from the communication nodes or the base stations plays crucial roles in offloading tasks to the datacentres, which is also affected by the current bandwidth availability, signal interception, intermittency [8] etc. Intermittency in the communication network creates dead-spots and coverage holes [14] in the MCC communication infrastructure, which in turn isolates the mobile terminals falling into such dead-spots and coverage holes. This scenario causes frequent suspensions of the mobile terminals from the cloud datacentres, which may cause resource scarcity in the mobile terminals and in turn affects the MCC privilege of anytime and anywhere computing. Location-aware [30] services facilitate the privilege of exploiting both the user mobility [31] and cloud resources whilst achieving energy efficiency.

Network latency is another dominating factor which has adverse effects on the energy consumptions of the communication infrastructures and also on the participating mobile and cloud datacentres. Latency has adverse effects on the energy consumptions when tasks are offloaded to distant datacentres and the presence of latencies [7], [32] in the communication channel degrades the crispness of the system response. In the MCC service infrastructure, even trivial applications involving user interactions incur communication delays imposed by latencies. Since users are acutely sensitive [33] to delays and jitter, the presence of latencies directly degrades both the QoS and QoE. The level of user tolerances decides the level of QoE depending on the nature of the applications. Such real-time tolerance levels of the task communication are usually measured [4] by Frames Per Second (FPS), with the acceptable lower bound FBS of 16 with no higher bound limits.

The energy costs [35] of communication usually depends on the bandwidth availability and the presence of latency, both play crucial roles in service delays. User perceived delays are actually an accumulation of process time and transfer time, where the transfer time is decided by the bandwidth availability and the presence of communication related latencies. Most often, cloud datacentres are built in less populated isolated places having low risk of disasters which would increase the distance of the mobile terminals from the datacentres. Interestingly, offloading and transfers using cellular networks consumes increased amounts of energy in comparison with the WLAN transfers. WLAN usually facilitates higher transmission bit-rate and thus enhances the transmission efficiency. In most of the existing MCC frameworks, the storage cost is often ignored since it is much lower than the processing costs. Since the storage capacities of the mobile devices are usually lower, tasks are also offloaded to the datacentres for storage space. This incurs additional storage costs [14] along with the communication and process costs.

Heterogeneity in the MCC infrastructure [4] is witnessed across the underlying hardware, platform, features, APIs etc., among the three integrated technologies. Heterogeneities in the mobile devices play vital roles in the mobile terminal services [36] such as data synchronisation, real-time push, and the mobile RPC (remote procedure call), channel management, resource discovery, securing authentication etc., among the mobile process clusters. The usual challenges imposed by the heterogeneities in the communication infrastructure arise due to the variations among the communication standards and protocols, nature and type of the wireless network, bandwidth and channel capacities etc. Custom built-in features, service policies, internal infrastructures, APIs, platforms, variations in the interoperable and portable frameworks are the usual exacerbating heterogeneities of the cloud counterpart. Such cross platform heterogeneities often causes issues [4] such as vendor lock-in, which is a state where the data, code, and comments cannot be easily transmitted from mobile to mobile, cloud to cloud, or mobile to cloud. The device and platform heterogeneity [37] also creates variations in measuring the performance of the execution, when the tasks are collaboratively executed among different devices, OS and platforms etc. Despite these inter- and intra-component heterogeneities, an effective MCC infrastructure should encompass the capacities of data distribution and process management between the mobile devices and the cloud datacentres. The inherent heterogeneity [38] among the integrated technologies can also be effectively utilised to achieve diverse user requirements, by the way of virtually homogenizing the core resources.

Context-aware data [39], [40] generally refers to the information about the environment and the situation, which can be utilised to characterise and model the situation of an entity (refers to users, scenario, and status of device, datacentres, and communication channel in the MCC infrastructure). The interactions between such entities are usually captured and processed for enabling a pervasive ubiquitous connectivity with a higher degree of synchronisation and data parallelism. User-centric context [41], [42] information helps

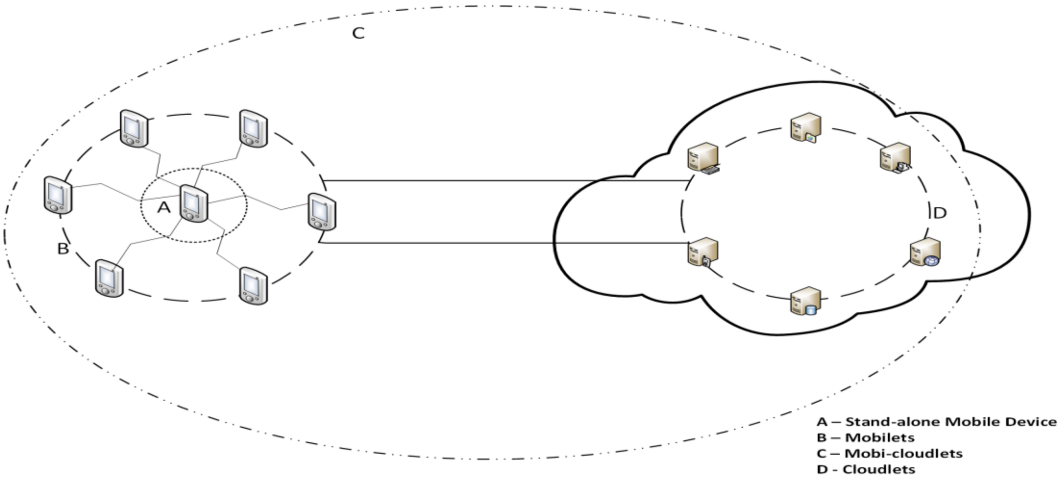


FIGURE 1. Mobilouds architecture.

supplementing user-centric computational capabilities to the process clusters, thereby resource scarcity can be avoided in the native terminals offering compute capabilities. But the exponential growth [43] and dynamic nature of such context-aware data and the drastic mobility of users in the recent days has imposed several challenges in processing such contextual data. Effective context-aware service strategies can help to dynamically adjust the service pattern in accordance with the user requirements in order to shorten the response times, identify resources at the mobile terminals, and to establish high-quality service networks etc. Alongside energy efficiency, congestion control and rate adaptations for interoperation are other aspects to consider whilst achieving convergence of the three core integrated technologies of MCC. Thus an effective MCC communication infrastructure requires high data rate, low latency, high mobility and high capacity network medium to enable energy efficient application offloading.

IV. MOBILOUDS FRAMEWORK

This section details our novel Mobilouds framework which is devised with the motivation of encouraging and extending the participation of the mobile devices in the collaborative execution between mobile terminals and the cloud datacentres.

A. MOBILOUDS ARCHITECTURE

The Mobilouds architecture is designed to reduce the excess energy consumption of the idle mobile terminal resources, by the way of utilising the mobile terminal resources in the actual task execution. Our proposed Mobilouds architecture encompasses four different process clusters comprising different compute capabilities, namely the native mobile terminal, mobilets, mobi-cloudlets and cloudlets, as shown in Fig. 1. The Mobilouds architecture encompasses both the real clouds and the virtual clouds. Real clouds are the typical process clusters in the traditional cloud datacentres. Virtual clouds are the miniature clouds formed of the nearby mobile resources in order to provide supplements for the source mobile terminal [5] for the purpose of executing light-weight computational workloads. Based on the application requirements and the resource availability, our Mobilouds architecture scales the appropriate process clusters for task execution, with the motivation of achieving increased utilisation of the available resources by minimising the idle resources whilst ensuring that the required amounts of resources are availed at an optimum level.

1) MOBILETS

Mobilets encompass a multi-tier ad-hoc clusters of mobile devices formed based on user population in the coverage area of the source mobile terminal requesting resources. The process capacities of the mobilet clusters depend on the available neighbour resources and user motivations to contribute resources to the mobilet clusters. Thus the mobilet process clusters are formed and runs on multiple mobile terminals and efficient enough to provide thread-safe services for enabling concurrent and parallel processing of the applications. 'Always-on' connectivity and 'on-demand' availability of the mobile resources are not always guaranteed from the nearby mobile devices to form the mobilet process clusters. Based on the user-provided resources, mobile terminals are invited by the source terminal to form the process clusters. Such mobilet clusters are suitable to process light-weight applications whose process logic can be complimented by the mobile terminals.

2) CLOUDLETS

In the case of heavy-weight applications, mobilets might not be suffice to process the application requirements. Majority or all of the process logic of such applications should take place in the cloud clusters. Such application tasks are offloaded to the cloud datacentres like a typical cloud service model. The process clusters are formed in the cloud resources called cloudlets and such clusters are referred to as [7] datacentre in a box.

3) HYBRID MOBI-CLOUDLETS

In some application executions, mobilets may not provide sufficient resources for entire processing but can still process the application requirements with the support services being provided by the cloud datacentres. Similarly, the heavyweight application process hosted on the cloudlets can be benefitted by the support services provided by the mobile terminals or mobilets. A hybridisation of the infrastructure-based and mobile ad-hoc based clusters with the combination of mobilets and cloudlets, named 'mobi-cloudlets' is devised in the Mobilouds architecture to form an effective MCC solution with the capacities of facilitating both the in-datacentre computing and mobile-cloud collaborative computing. This hybridised mobi-cloudlets are able to utilise both the cloud resources and the mobile terminal resources collaboratively for energy efficient computing. Such a mobi-cloudlet architecture can be effectively managed by a distributed management system, rather than a centralised controller. Telecom grade clouds [32] can be created in such distributed Mobilouds architecture to host the server modules in the mobile terminals, by the way of effectively utilising the network components. A hybridised infrastructure encompassing mobilets, cloudlets and mobi-cloudlets can be established through access networks. All the mobilets, cloudlets and mobi-cloudlets process clusters are subjected to pre-use customisation and post-use cleanup which ensures effective proximities of the MCC elements.

B. MOBILOUDS APPLICATION STACK

Further to the Mobilouds architecture, we devise a novel Mobiloud application to support the decision making and auto-scaling of the process clusters in the Mobilouds architecture. The mobiloud application is a software process deployed on each participating mobile device. The application consists of two major inseparable components which ensure that devices are actively participating as both service consumer and service provider and not simply service leeching.

The Mobilouds application stack is a master-slave configuration, each device has a single master process and multiple slave processes. The Master process is responsible for locating processing capability and requesting use. Each mobile device has a single master process which prevents a single device from consuming all other mobilet resources for multiple jobs. The Mobiloud application stack is illustrated in an application scenario of diverse resource availability in Fig. 2.

The resource availability in a given mobile terminal can be defined as a set of slave slots with profiles S in relation to the corresponding available time profile T. Every individual slave slot profile s_i contained in S will be governed by the functions c_a, s_d, m_a, p_d, t_{ai} where c_a is the number of processing slots offered by a given mobile terminal, s_d is the static processor speed of that device, m_a is the RAM space available in that device, p_d is the remaining battery power in that device and t_i is the time slot available in that device for the source terminal to consume. Thus slave slot profiles in a given mobile terminal can be defined as in (1) to (3).

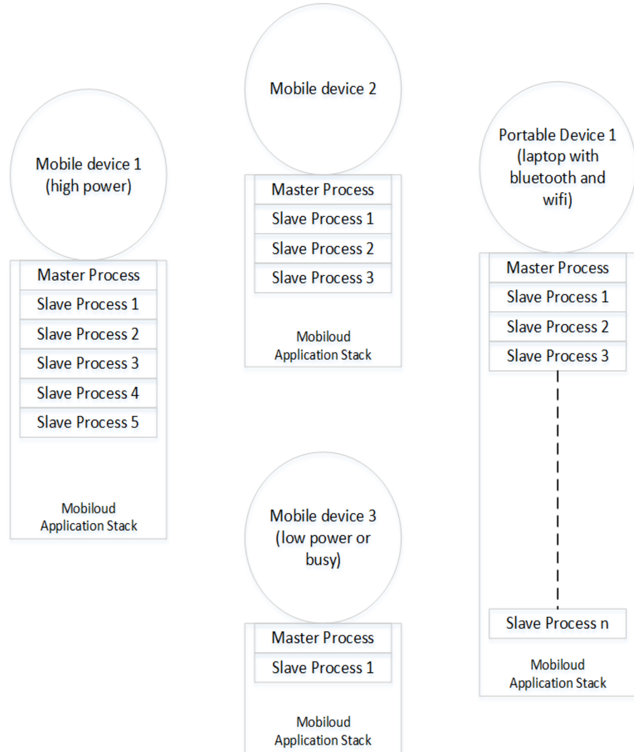


FIGURE 2. **Mobilouds application stack.**

$$S = \{s_1, s_2, s_3, \ldots\ldots s_n\} \tag{1}$$

$$T = \{t_1, t_2, t_3, \ldots\ldots t_n\} \tag{2}$$

$$s_i = \{f(c_a, m_a), f(s_d), f(p_d), f(t_i)\} \tag{3}$$

Since the slave slots evolves in time in spite of terminals both contributing and consuming resources, we define an n-step transition probability for the slave slots. So the availability status of the slave slots at a given time is given by (4).

$$s_j^{(t+1)} = \sum\nolimits_{i=1}^{n} s_i^t s_{ij} \tag{4}$$

where s_i^t is the status of the i^{th} slave slot at time t_i, s_{ij} is the transition probability of the slave slot from time t_i to $t_{(i+1)}$.

In general, the task execution should be transferred to a higher process cluster from the current process cluster during resource scarcity by a rapid, invisible and seamless hand-off in an effective MCC framework. But such handoffs usually incur additional delays resulting from service initiations whilst inviting resources. The physical proximity of the Mobilouds is crucial for effective collaborative processing, and the end-to-end response time of application execution [7] must be fast and predictable. Usually, users cannot expect constant turn-around times in MCC service deliveries due to several dynamic external factors. Effective decision making for allocating the appropriate process clusters in the Mobilouds architecture is achieved by evaluating the current situation and the nature of the application requirements.

The Mobliouds application stack at the source terminal runs this initial estimation before initiating the cluster formation. The concept behind this effective decision making for selecting the optimum level of process cluster is detailed in the following section.

V. CLUSTER SCALING

Utilising the various process clusters of the Mobilouds architecture by reducing the idle resources is challenging since both the task load and the capacity of the process clusters evolve over time. Mobilouds architecture supports hierarchical evolution of the process cluster from a stand-alone mobile device to a multi-tier process cluster. But this scaling should be lined in accordance with both the currently available resources and the nature and requirements of the applications to be processed. The Mobilouds architecture can both be upgraded to a higher-end process cluster and also be downgraded to reduce the presence of idle resources. To achieve an optimum scaling of the cluster resources, we define two major classes for the applications waiting to be processed. The first class is where the application or data is stored totally or substantially on the local mobile device. The immediate implication is that storage space is not an issue. The second class is where the application or data is totally or significantly remote from the local mobile device. The most significant implication is that the data may need to be transferred to the mobile device before use.

Based on the two classes of application, the resource requirements for the successful execution of any tasks at a given time can be defined as the user requirement profile u_i containing the composite $\{c_r, m_r, t_r\}$ where c_r is the required amount of CPU in terms of the number of processing slots required, and m_r is the required amount of memory, and t_{ri} is the time scale allowed to process the tasks based on the user expectation, shown in (5).

$$u_i = \{f\ (c_r), f(m_r)\ , f\ (t_r)\} \tag{5}$$

The user expectation for executing an application can thus be defined as the probability of user expectations $P\,(u_i)$ in condition with the probability of the slave slot status $P(s_j)$, as in (6).

$$E\,(u_i) = u_i\left(P\,(u_i)\,|P(s_j)\right) \tag{6}$$

The number of processing slots required to execute any tasks is computed by balancing the trade-off between energy and core utilisation whilst completing the tasks in the preferred amount of time. For instance, a given application can either be completed with substantial usage of resources in a real quick time or with a reasonable usage of resources with increased processing time. So, the QoE of the users in terms of user tolerances is used as a deciding factor for choosing the required number of processing slots and the time within which the process should be completed.

Computing the required amounts of CPU and memory alongside the allowed timeslot allows more flexibility in forming the optimum process cluster. Upon resource request, the source terminal receives the composite S, shown in (3). Upon receiving this response from the devices in the coverage area of the source terminal, the source terminal selects the devices depending on the resource availed by them and invites them to form the process cluster. The number of devices availing resources depends on the number of active devices in the coverage area of the source terminal and the state of their resource availability. The required amounts of CPU and memory are contrasted against the resources availed and the decision is made up on whether to use the availed process cluster or to request a more capable process cluster.

This initial estimation of contrasting the required resources against the available resources is used to decide up on upgrading and downgrading the current process cluster for reducing both the excess energy consumption and idle resource in the process cluster. Now, we further introduce three different use cases of the required and the available amounts of resources in order to select the optimum level of process cluster in the Mobilouds architecture, assuming that the execution starts in the native mobile device.

A. CASE (i) RESOURCE REQUIREMENTS LESS THAN RESOURCE AVAILABLE

With both the required amounts of CPU and memory being less than the available resources in the mobile device, the mobile terminal resources should be sufficiently capable to execute the application requirements. No more actions are required in this case, since there is no possibility of reducing any excess utilisation of resources and the application is executed in the native mobile device.

B. CASE (ii) RESOURCE REQUIREMENTS GREATER THAN RESOURCE AVAILABLE

In this process scenario, the mobile device resources cannot satisfy the requirements of the task execution, but the source terminal may still host some parts of the application execution depending on the class of the application. Now, the mobile device triggers the initiation of the mobilets. If the desired resource of CPU/memory is available in the mobilets coverage area, nearby devices are invited to form the mobilet process clusters. Upon accepting this invitation, mobilets are formed through the available communication medium and the application is executed. Here the process clusters are upgraded from a stand-alone mobile device to the mobilet process cluster. If the requested resources are not available in the coverage area of the native mobile device, then the corresponding tasks (CPU/memory) are offloaded to the distant cloud datacentre. If the resources are partially available but still cannot completely satisfy the application requirements, mobilets are still formed to host a part of the application execution and additional resources are requested from the cloud datacentre by which the process cluster evolves into mobi-cloudlets. The involvement of the mobile terminals in the mobi-cloudlets will be decided based on the computation intensity of the application. For high computational applications, the tasks are offloaded to the cloud datacenters and the mobile terminals involved in the mobi-cloudlets are released.

C. CASE (iii) RESOURCE AVAILABLE EXCEEDING RESOURCE REQUIREMENTS

The mobile terminals involved in the process clusters will update their status of resource availability in a timely fashion since the available amounts of resources evolve over time. If the available amounts of resources drops below the actual requirements, additional resources are requested to join the clusters thereby upgrading the current cluster. But, if there are excess amounts of resources being idle in the process clusters then such terminals with excess resources are released from the process cluster and the cluster resources are downgraded accordingly in order to reduce the resources being idle in the execution process. Such released terminals are refreshed and can contribute resources to other peer source terminals requesting resources. In this way, the cloudlets are downgraded to mobi-cloudlets, and mobi-cloudlets to mobilets, and mobilets to stand-alone mobile device accordingly based on the actual scenario and the status of the available resources.

Assuming that the tasks are initially executed in the stand-alone mobile devices, we introduce a decision making problem under uncertainty based on the above three cases. With the Mobilouds architecture evolving in a hierarchical fashion, selecting the optimum level of process cluster is defined as a sequential decision making problem. Now the optimum cluster can be formed by either upgrading, downgrading or staying at the current level of process cluster with the most energy-time balancing solution is availed first in the sequential order based on Hurwicz criterion, as shown in (7).

$$f(l) = \alpha\,(cluster1) + (1-\alpha)\,(cluster2) + (2-\alpha)\,(cluster3) \tag{7}$$

where $f(l)$ gives the energy-time balanced process cluster and α is the energy efficient weight parameter obtained from the initial resource estimation performed by the source terminal. This weight parameter is chosen in such a way to balance the trade-off between energy consumption and job execution duration. The process of optimum selection of the communication medium and achieving the most energy efficient process cluster for collaborative MCC execution is explained in the following section.

VI. MOBILOUDS IMPLEMENTATION

Upon the source terminal triggering the resource request, the Master will search for process capability within the local device and use this whenever possible. If the local resource is expended or determined to be unsuitable, the Master process

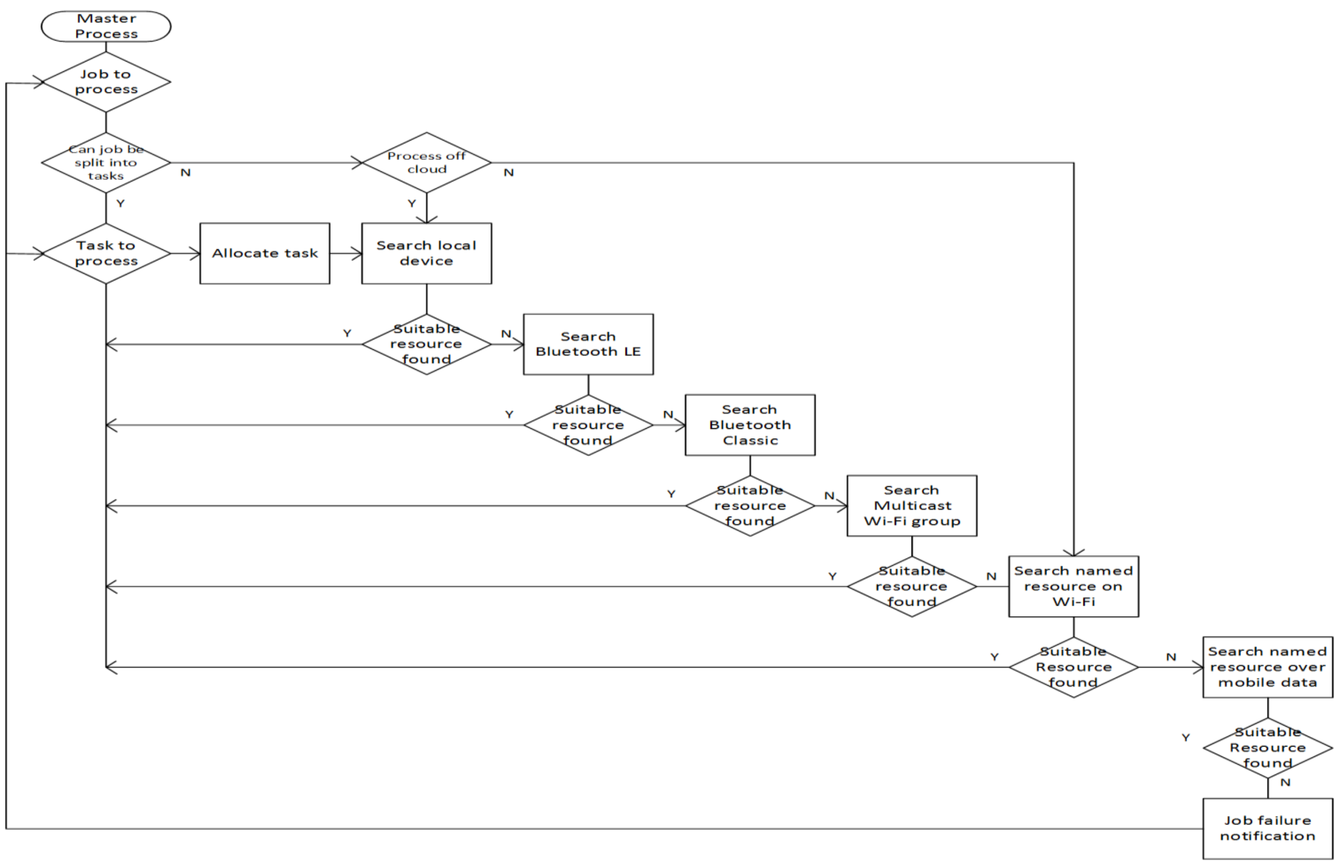


FIGURE 3. Mobilouds master process.

searches for nearby resource and triggers a service request. The Master uses Bluetooth Low Energy locality first, followed by classic Bluetooth, local Wi-Fi (multicast group), Wi-Fi and finally cellular networks in the order of 5/4/3G. This sequential order of communication medium is employed to achieve the most appropriate solution available, given the preference of the least energy consuming communication medium is selected first.

The Master process is periodic in nature, resources requirements that extend beyond the "contract time" have to be requested again after the contract expires. Resources are considered suitable if they include processing power, memory space, and electrical power, and are available for a minimum period capable of fulfilling the proposed task within a small multiple of the contract time. The Slave process is responsible for identifying and providing processing capability. The Slave process periodically tests the local device for resources and creates a standard slave processing slot based on the slave slot profile S. The slave process creates multiple slots when possible and makes them available to a Master process enquiry, and slots are not actively advertised.

The Slave process will respond to Bluetooth service enquiries and to multicast group requests on either ipv4 or ipv6 Wi-Fi only. It will reply only if it has resource slots capable of fulfilling the task. Fixed devices running the Mobiloud mobile process stack may also respond to local requests and would be preferred over local mobile devices. The periodic nature of the application ensures that no task can consume all of the resources of a device over a long period and provides an automatic contract release in the event of a communication failure.

When there are no suitable local processing slots available, the Master process will request service from a named external service. The lack of suitable processes may be due to there being no other devices in the local vicinity which are running Mobiloud application, it may also be due to resource depletion on all other local Mobiloud enabled devices or that the resource required is beyond the capability of the mobile devices. The use of named services means that the services must be advertising or have a DNS entry. Advertising is not part of the Mobiloud process for mobile devices and therefore the service must be provided by a fixed device. The choice of servicing device is hierarchical from the user configurable resolvable name and from device handoff to cloud services. The logical functioning of the Master process in the Mobilouds application is illustrated in Fig. 3.

In general master slave architectures do not scale well with MCC process clusters, if the slaves do not have sufficient privileges in the collaborative task execution. The slaves in the process clusters should be able to transform into a master with preferred privileges, in order to have an uninterrupted task execution. Our proposed Mobilouds application facilitates controlled privileges to the slaves in the application stack, by which the slaves can decide their level of resource contribution based on the current state of their resource utilisation. Fig. 4 illustrates the functionalities of the slaves

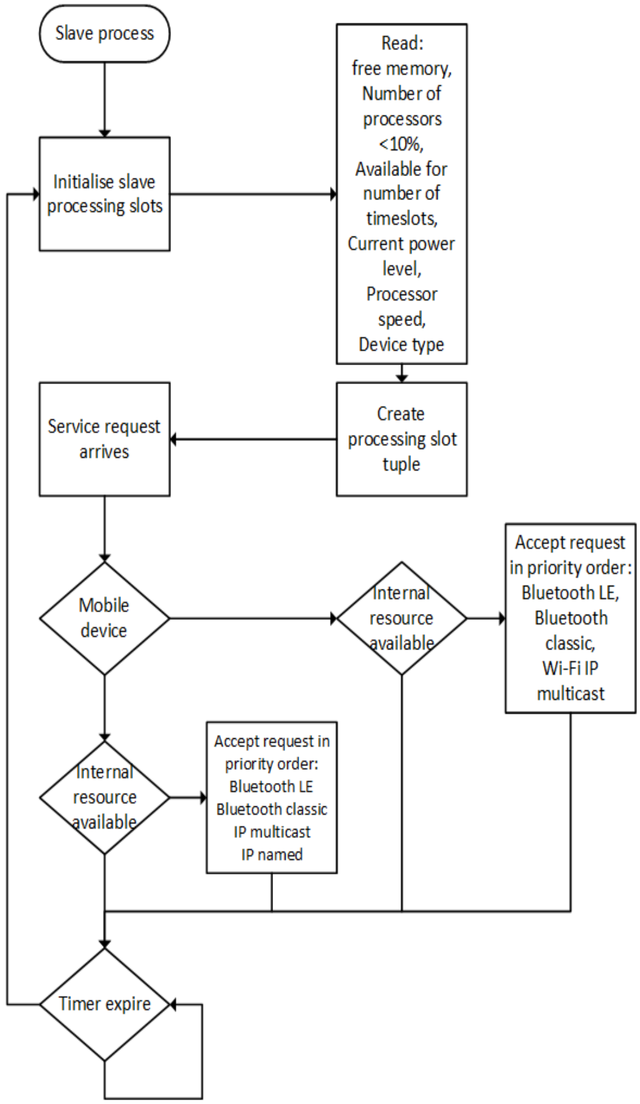


FIGURE 4. **Mobilouds slave process.**

in the MCC collaborative execution of Mobilouds. Fig. 5 describes the evolution process algorithm of the Mobilouds architecture.

The Mobilouds architecture evolves from the stand alone mobile terminals in a hierarchical fashion based on the resource requirements and availability. The Mobilouds application employs a bottom-up approach whilst facing resource constraints and a top-down approach when there are surplus resources staying idle in the process cluster. Upon the availability of several process clusters, the process logic of selecting the optimum process cluster is shown in Fig. 6.

VII. PERFORMANCE EVALUATION

This section demonstrates the operational efficiency of Mobilouds application in terms of reducing the energy consumption whilst executing tasks in the Mobilouds architecture in a collaborative fashion, based on a pre-defined use case scenario. Fig. 7 illustrates the currently available

TABLE 1. Summary of resource availability.

	Access Type	Access Speed	Memory Size	Process Speed	Process Power
Local Mobile (Android)	n/a	n/a	1.50E+06	2.00E+04	20 mW
Mobile 1 (Android)	Bluetooth Low Energy	1.00E+06	1.00E+06	1.00E+04	20 mW
Laptop 1 (i7-4770T)	Bluetooth	3.00E+06	3.00E+06	2.00E+06	52.5 W
Laptop 2 (i7-4770T)	Wi-Fi	2.00E+08	1.00E+07	2.00E+06	52.5 W
Local Server (Xeon E3-1220v5)	Wi-Fi	2.00E+08	1.00E+09	3.50E+07	200 W
Cloud Server (Xeon E5-2699v3)	4G	1.00E+08	1.00E+10	3.50E+07	360 W
Cloud Server (Xeon E5-2699v3)	5G	1.00E+09	1.00E+10	3.50E+07	360 W

Input: $\{c_r, m_r, t_r\}$
Output: $\{c_a, s_d, m_a, p_d, t_a\}$,
If case (i)
1. Execute tasks in the mobile device

If case (ii)
2. Trigger Resource Request
3. Select the communication medium in sequential order
   (Bluetooth LE, Bluetooth Classic, Multicast Wi-Fi, Named Wi-Fi, Cellular Network 5G/4G/3G)

If Suitable Resource Found
4. Invite the resource to join the process cluster

Else
5. Go to Step 2 and upgrade the communication medium by one tier
   The process cluster evolves into mobilets, mobi-cloudlets and cloudlet accordingly upon every iteration

If case (iii)
6. Release resources to downgrade the process cluster

FIGURE 5. Cluster evolution algorithm.

1. Use local resource native to the user
2. If native resources not capable, use local mobiles
3. If local mobiles not capable, use local Bluetooth laptop then Wi-Fi laptops
4. If local laptops not capable, use Wi-Fi local server on site
5. If local serer not capable, use internet server by 5G/4G/3G (based on availability)

FIGURE 6. Cluster selection logic.

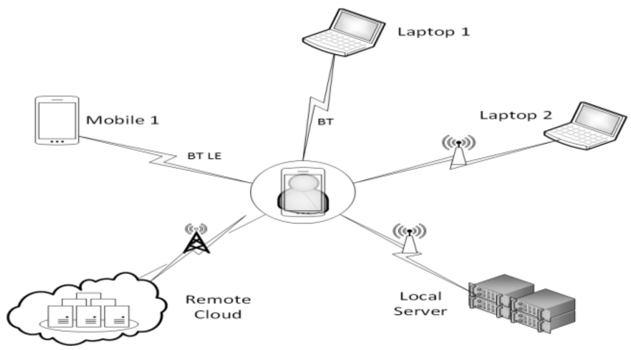


FIGURE 7. Evaluation scenario.

TABLE 2. Access technology considerations.

Access Technology	Transmission Power
Bluetooth Low Energy	0.01
Bluetooth	0.1
Wi-Fi	0.5
Fast Ethernet LAN	22.5
Fast Ethernet WAN	150
Gigabit Ethernet LAN	22.5
Gigabit Ethernet WAN	150
4G	1
5G	0.75

resources and their corresponding available communication medium in the Mobilouds architecture. Table 1 depicts the currently available process slots availed by the resources. Our performance evaluations are based on real-life standards and metrics [44]–[48] used in MCC infrastructure. Table 2 summarizes our considerations during this performance evaluation. Further, we consider that there is no process failure or communication failure in this evaluation.

Based on the scenario presented in Fig. 7, we evaluate the efficiency of the Mobilouds application in processing a job consisting of 5 tasks of different sizes as 100kByte, 1MByte, 5MByte, 100Mbyte and 2MByte, named task 1 through to task 5 accordingly. With the local android mobile device being less capable than the actual process requirements in terms of process speed and available slave slots, it triggers the resource request. Now the objective of the Mobilouds application is to choose the most energy and time efficient process cluster from the available resources. Since, energy and time parameters are dependent on each other, it is always optimum to compute the energy-time trade-off for choosing the optimum process cluster based on the application requirements. In order to evaluate the most energy and time efficient cluster, Mobilouds runs an initial estimation of the available resources (according to the evaluation scenario) based on the job requirements. This initial estimation is to compute the process time and process power anticipated on the resources available in the cluster based on the process slots availed by the participating resources. Table 3 through to Table 7 depicts this initial estimation summary of processing the individual

TABLE 3. Task 1 process estimation summary.

Device	Process Time				Process Power					
	Transit Out	Process	Transit In	Total Time	Transmit	Transit	Process	Return	Receive	Total Power
Local	n/a	5.00E+00	n/a	5.0	n/a	n/a	1.00E-01	n/a	n/a	0.1000
Mobile 1	8.00E-01	1.00E+01	8.00E-01	11.6	8.00E-03	n/a	2.00E-01	n/a	8.00E-03	0.2160
Laptop 1	2.67E-01	5.00E-02	2.67E-01	0.583	2.67E-02	n/a	2.63E+00	n/a	2.67E-02	2.6783
Laptop 2	4.00E-03	5.00E-02	4.00E-03	0.058	2.00E-03	2.00E-03	2.63E+00	2.00E-03	2.00E-03	2.6330
Local server	4.00E-03	2.86E-03	4.00E-03	0.0108	2.00E-03	2.00E-03	5.71E-01	2.00E-03	2.00E-03	0.5794
Cloud 4G	8.00E-03	2.86E-03	8.00E-03	0.0188	8.00E-03	4.00E-03	1.03E+00	4.00E-03	8.00E-03	1.0526
Cloud 5G	8.00E-04	2.86E-03	8.00E-04	0.0044	6.00E-04	4.00E-04	1.03E+00	4.00E-04	6.00E-04	1.0306

TABLE 4. Task 2 process estimation summary.

Device	Process Time				Process Power					
	Transit Out	Process	Transit In	Total Time	Transmit	Transit	Process	Return	Receive	Total Power
Local	n/a	5.00E+01	n/a	50.000	n/a	n/a	1.00E+00	n/a	n/a	1.000
Mobile	8.00E+0	1.00E+02	8.00E+00	116.00	8.00E-02	n/a	2.00E+00	n/a	8.00E-02	2.160
Laptop 1	2.67E+0	5.00E-01	2.67E+00	5.8333	2.67E-01	n/a	2.63E+01	n/a	2.67E-01	26.78
Laptop 2	4.00E-02	5.00E-01	4.00E-02	0.5800	2.00E-02	2.00E-02	2.63E+01	2.00E-02	2.00E-02	26.33
Local server	4.00E-02	2.86E-02	4.00E-02	0.1085	2.00E-02	2.00E-02	5.71E+00	2.00E-02	2.00E-02	5.794
Cloud 4G	8.00E-02	2.86E-02	8.00E-02	0.1885	8.00E-02	4.00E-02	1.03E+01	4.00E-02	8.00E-02	10.52
Cloud 5G	8.00E-03	2.86E-02	8.00E-03	0.0445	6.00E-03	4.00E-03	1.03E+01	4.00E-03	6.00E-03	10.30

TABLE 5. Task 3 process estimation summary.

Device	Process Time				Process Power					
	Transit Out	Process	Transit In	Total Time	Transmit	Transit	Process	Return	Receive	Total Power
Local	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Mobile 1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Laptop 1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Laptop 2	2.00E-01	2.50E+00	2.00E-01	2.900	1.00E-01	1.00E-01	1.31E+02	1.00E-01	1.00E-01	131.65
Local server	2.00E-01	1.43E-01	2.00E-01	0.5428	1.00E-01	1.00E-01	2.86E+01	1.00E-01	1.00E-01	28.9714
Cloud 4G	4.00E-01	1.43E-01	4.00E-01	0.9428	4.00E-01	2.00E-01	5.14E+01	2.00E-01	4.00E-01	52.6286
Cloud 5G	4.00E-02	1.43E-01	4.00E-02	0.2228	3.00E-02	2.00E-02	5.14E+01	2.00E-02	3.00E-02	51.5286

TABLE 6. Task 4 process estimation summary.

Device	Process Time				Process Power					
	Transit Out	Process	Transit In	Total Time	Transmit	Transit	Process	Return	Receive	Total Power
Local	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Mobile 1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Laptop 1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Laptop 2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Local server	4.00E+00	2.86E+00	4.00E+00	10.857	2.00E+00	2.00E+00	5.71E+02	2.00E+00	2.00E+00	579.42
Cloud 4G	8.00E+00	2.86E+00	8.00E+00	18.857	8.00E+00	4.00E+00	1.03E+03	4.00E+00	8.00E+00	1052.5
Cloud 5G	8.00E-01	2.86E+00	8.00E-01	4.4571	6.00E-01	4.00E-01	1.03E+03	4.00E-01	6.00E-01	1030.5

tasks in all the available resources. Fig. 8 illustrates the anticipated process time and process power consumption whilst executing the tasks in the available resources individually Tables 4–7. From Fig. 8, 5G Cloud offers the lowest process time for all the five tasks when processing the tasks individually. The native local mobile resource offers the lowest process power for tasks 1 and 2, and the local server offers the lowest process power for tasks 3, 4, and 5 respectively. Given that the job is already divided into task threads in the native mobile, further allocation of the individual tasks in different threads is not feasible. Each task must therefore be processed by a single resource.

TABLE 7. Task 5 process estimation summary.

Device	Process Time				Process Power					
	Transit Out	Process	Transit In	Total Time	Transmit	Transit	Process	Return	Receive	Total Power
Local	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Mobile 1	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Laptop 1	5.33E+00	1.00E+00	5.33E+00	11.666	5.33E-01	n/a	5.25E+01	n/a	5.33E-01	53.566
Laptop 2	8.00E-02	1.00E+00	8.00E-02	1.1600	4.00E-02	4.00E-02	5.25E+01	4.00E-02	4.00E-02	52.660
Local server	8.00E-02	5.71E-02	8.00E-02	0.2171	4.00E-02	4.00E-02	1.14E+01	4.00E-02	4.00E-02	11.588
Cloud 4G	1.60E-01	5.71E-02	1.60E-01	0.3771	1.60E-01	8.00E-02	2.06E+01	8.00E-02	1.60E-01	21.0514
Cloud 5G	1.60E-02	5.71E-02	1.60E-02	0.0731	1.20E-02	8.00E-03	2.06E+01	8.00E-03	1.20E-02	20.6114

TABLE 8. Availed mobilouds process clusters.

Process Cluster	Use Case	Process Time	Process Power
Mobilouds 1 (T1 - local; T2- laptop 1; T3, T4, T5 – Local Server)	Resources available locally.	11.61	646.86
Mobilouds 2 (T1 - local; T2- laptop 1; T3 – Local Server, T4, T5- Cloud 5G)	Local server not completely available for T4. 5 G Cellular network available.	5.83	1107.037
Mobilouds 3 (T1 - local; T2- laptop 1; T3, T5 – Local Server, T4- Cloud 4G)	Local server not completely available for T4. 4 G Cellular network available.	18.85	1119.92

The primary objective of Mobilouds can be viewed from two perspectives. First is to achieve the lowest time and power consuming process cluster for the entire job. Secondly, the idle resources of the native and local resources should be effectively utilised in processing the job to reduce the resource idleness and corresponding power drains. Now, Mobilouds application offers the optimum process cluster by optimising both the process time and process power. From the given process scenario, Mobilouds application offers three different process clusters as shown in Table 8 in accordance with the process logic shown in Fig. 6. The process clusters are formed in such a way that the idle resources of the local devices are effectively utilised to process smaller tasks, whilst offloading heavy tasks to the cloud servers.

From Table 8, Mobilouds 1 offers the best trade-off between time and energy and Mobilouds 2 offers the lowest possible process time. When there is resources scarcity in the local devices, Mobilouds application chooses Mobilouds 2 cluster upon the availability of 5G cellular network. Otherwise, Mobilouds 3 is formed using 4G cellular network to offload tasks to the distant datacentre. We further evaluate the efficiency of Mobilouds by comparing the time and power efficiencies of the Mobilouds process clusters against offloading and processing the entire job in the distant datacentre using both 4G and 5G cellular networks.

Fig. 9 illustrates the process time and process power of the three Mobilouds clusters alongside a complete cloud solution using 4G and 5G cellular networks respectively. From Fig. 9, it is clear that a complete 5G cloud solution for the entire job offers the lowest time, followed by Mobilouds 2, Mobilouds 1, Mobilouds 3, and the 4G cloud solution respectively. Again, Mobilouds 2 uses the 5G cloud solution for heavy tasks, thus it is time efficient. It is worthy of note that a complete cloud solution always incurs the non-negligible idle resources of the native devices staying active and therefore experiencing undesirable power consumption whilst waiting for the cloud solution. Clearly Mobilouds 1 is the most energy efficient process cluster, which involves processing the job locally using the mobile, laptop and the local server. Though three different cluster combinations are provided by the Mobilouds application, the optimum cluster should be chosen based on the requirements and outcomes of individual scenarios. Despite the time and energy costs incurred in the Mobilouds 2 and 3, the computational costs of the cloud solutions cannot be avoided in the two cloud assisted solutions. This additional computational cost is avoided in Mobilouds 1 due to the user-owned resources. Thus Mobilouds 1 is not only efficient in optimising the time-energy trade-off, but is also cost effective.

VIII. APPLICATION OF THE WORK

This section explains the practical applicability of Mobilouds in [49] Satellite Navigation application. Consider a scenario where a user needs to travel from Derby to London in the United Kingdom on road. The road distance between Derby to London is 130 miles and a car journey at average speed should take approximately 2 hours and 30 minutes. GPS can be used to determine the most time efficient route based on distance travelled and route congestion and therefore avoiding unnecessary delays. Given the fact that traffic density varies with time, predetermination of the entire end-to-end route is unlikely to be ideal; dynamic estimation of the effect of temporal factors along the route may suggest alternative choices as the journey progresses. The entire route can be

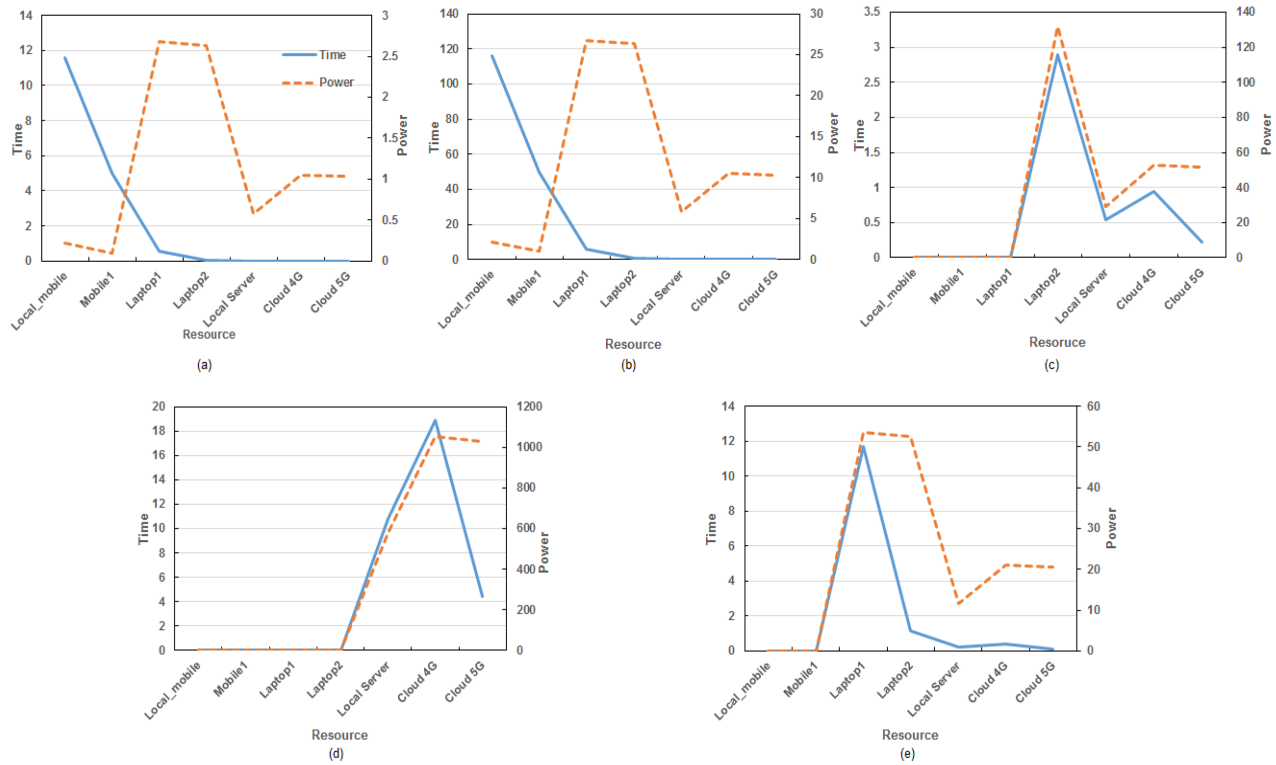


FIGURE 8. Energy and time consumption of task execution (a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4 (e) Task 5.

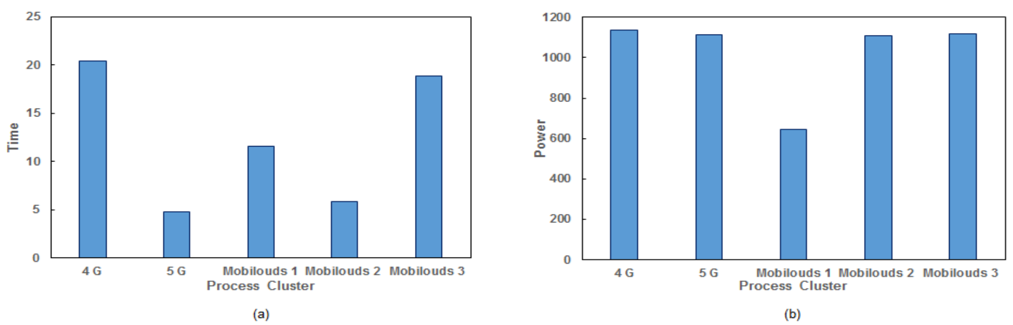


FIGURE 9. Process efficiency (a) task duration (b) power consumption.

considered in at least three sections: exiting Derby, trunk routes from Derby to London and traversing London to the end destination. The most challenging parts of the route determination will be the urban road networks of Derby and London as to offer the highest diversity of routes with the most dynamic traffic flows. Route determination can be split into a number of tasks, determining the Derby city traffic first, followed by the motorway and the London city traffic at much later stage of the journey. The entire journey requires the road map with all current and up to date traffic information on it. But having the entire map at the time of setting off from Derby does not help all the way through London.

A Mobiloud application would split the job into tasks of determining the traffic map and downloading at various stages of the journey. The mobile device would initially download only the Derby street map from a cloud application, while the cloud determines and calculates a general route based on existing information. This saves storage space and processing capability on the local device while ensuring that the map data is totally current. It also allows the journey to begin before the full route has been determined. Mobilets formed of the available nearby devices can provide "local knowledge" of the current Derby City traffic, the location and local data is then processed by the mobile device. As the mobile device approaches the Derby Arterial network, a request is made for the next stage of the journey from the cloud service. The cloud service has been able to process current road and traffic conditions without any processing or communication cost to the mobile device. The device is then provided with map data related to the trunk section of the journey. The cloud service continues to process road and traffic data, the local device can collect route information.

In the event of a major change of circumstance or after a period of time, and the local device can send and receive updates. Finally, as the device approaches the London arterial road network, a request is made to the cloud service for the map and traffic data for the terminal leg of the journey.

The communication method during the entire journey may have changed several times, for example, if the journey began at a home location local Wi-Fi may have been available; mobilets would have formed over Bluetooth; the trunk journey would have used cellular data (5/4/3G) and the London section may have been able to access local hotspot Wi-Fi and Bluetooth. Processing geographically local information using the resources of the mobile device and geographically remote data in cloud service encourages collaborative participation. Sharing information regarding local traffic conditions is mutually beneficial to all mobile service users actively promoting the MCC collaborative framework.

IX. FUTURE WORK

Our future work will span across the following research gaps in the MCC processing infrastructure.

A. TASK DISTRIBUTION AND SCHEDULING

An autonomous task distribution and scheduling algorithm will be devised between the mobile devices and the cloud for the purposes of achieving effective collaboration and synchronization in the process cluster. The feasibilities of employing GPGPUs in mobile devices for developing an autonomous task distributor for scheduling tasks among the various levels of process clusters will be explored.

B. LOAD BALANCING

A load balancing mechanism for optimum scaling of the process loads across the mobile devices participating in the process cluster will be investigated with the motivation of auto-initialisation of new process clusters when the current clusters are overloaded based on overload thresholds in individual mobile devices.

C. REUSABILITY

The reusability and extensibility of the mobile resources in minimum time interval between successive task executions will be investigated.

D. MIDDLEWARE

The development of a management layer with higher degree of coupling between the functional code of the diverse mobile devices beyond their heterogeneities in platform, OS etc., and the cloud datacentre heterogeneity will be investigated.

E. FAULT TOLERANCE

The potential of utilising the mobile resources and extending their participation in the collaborative MCC execution even when the mobile devices face disconnections from the process clusters will be explored.

X. CONCLUSION

Energy efficiency is increasing in demand in any form of high performance computing. This paper addresses the issues of idle resource energy implications in the MCC infrastructure and further investigates the possibilities of reducing the presence of such idle resources, thereby reducing the overall energy consumption levels whilst executing jobs in a collaborative fashion between the mobile terminals and the cloud datacentres. Energy implications in the MCC collaborative infrastructures are witnessed across all the underlying integrated technologies. Though the energy consumption levels of the idle mobile terminals are comparatively insignificant to those of the communication and cloud counterparts, such energy implications are non-negligible in terms of the mobile standards and capacities causing undesirable battery drains.

The efficiencies of communication infrastructures are evolving quickly, with 4G cellular standards are deployed in most of the developing countries and 5G deployments are not far off. Though the energy and performance efficiencies of the cellular networks are evolving at a rapid pace, their deployment efficiencies depends on the way of their utilisation and implementation strategies. Achieving energy efficiency in cloud datacentres is always monumental, since datacentres incur energy consumptions across a multi-dimensional components including operating servers, lighting and cooling systems, switches etc., alongside the actual process energy.

Mobilouds architecture and its application process provides a complete energy solution by the way of achieving the least possible energy efficient solution across the integrated technologies. Our performance evaluations demonstrate that the offloading and process solution of Mobilouds achieves an optimum trade-off between energy and time. Clearly, offloading and processing the job requirements in the distance datacentre using 5G cellular networks incur the least processing time and less energy consumption to those of using a 4G cellular network. This leads us to infer that 5G deployments would certainly be time and energy efficient. However, this is subjected to the availability of the 5G cellular network and the capacities of the mobile devices participating in the job execution.

Rather than a single solution agenda, Mobilouds offer the maximum number of possible solutions to the users based on the actual availability of the resources with a different choice of energy-time trade-off. This provides users with an increased flexibility to choose the optimum solution based on the real-time requirement and the nature of the job applications. Our performance evaluations prove that Mobilouds is effective in achieving the optimum trade-off between process time and energy and in reducing the undesirable energy consumptions across all the three underlying process components of the MCC process infrastructure.

REFERENCES

[1] A. Alzahrani, N. Alalwan, and M. Sarrab, "Mobile cloud computing: Advantage, disadvantage and open challenge," in *Proc. 7th Eur. Amer. Conf. Telemat. Inf. Syst.*, Valparaiso, Chile, 2014, Art. no. 21.

[2] G. Mathew and Z. Obradovic, "Improving computational efficiency for personalized medical applications in mobile cloud computing environment," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Philadelphia, PA, USA, Sep. 2013, pp. 535–540.

[3] S. Kitanov and T. Janevski, "State of the art: Mobile cloud computing," in *Proc. 6th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Tetova, Macedonia, May 2014, pp. 153–158.

[4] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 369–392, 1st Quart., 2014.

[5] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.

[6] T. Lose and M. Thinyane, "A cloud computing platform to augment mobile phone use in marginalized rural areas," in *Proc. SATNAC*, Western Cape, South Africa, 2012, pp. 1–2.

[7] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[8] S. S. Qureshi, T. Ahmad, K. Rafique, and Shuja-ul-islam, "Mobile cloud computing as future for mobile applications—Implementation methods and challenging issues," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Beijing, China, Sep. 2011, pp. 467–471.

[9] A. Ravi and S. K. Peddoju, "Energy efficient seamless service provisioning in mobile cloud computing," in *Proc. IEEE 7th Int. Symp. Service-Oriented Syst. Eng.*, Redwood City, CA, USA, Mar. 2013, pp. 463–471.

[10] J. Li *et al.*, "CyberGuarder: A virtualization security assurance architecture for green cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 2, pp. 379–390, Feb. 2012.

[11] H. Al-Aqrabi, L. Liu, R. Hill, and N. Antonopoulos, "Cloud BI: Future of business intelligence in the cloud," *J. Comput. Syst. Sci.*, vol. 81, no. 1, pp. 85–96, Feb. 2015.

[12] B. Li, J. Li, and L. Liu, "CloudMon: A resource-efficient IaaS cloud monitoring system based on networked intrusion detection system virtual appliances," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 8, pp. 1861–1885, Jun. 2015.

[13] W. Lee, J. Jung, and H. Kim, "Analyzing extent and influence of mobile device's participation in mobile cloud computing," in *Proc. Int. Conf. ICT Converg. (ICTC)*, Jeju, South Korea, Oct. 2013, pp. 767–772.

[14] X. Liu, C. Yuan, Z. Yang, and Z. Hu, "An energy saving algorithm based on user-provided resources in mobile cloud computing," in *Proc. IEEE 78th Veh. Technol. Conf. (VTC Fall)*, Las Vegas, NV, USA, Sep. 2013, pp. 1–5.

[15] M. Nir, A. Matrawy, and M. St-Hilaire, "An energy optimizing scheduler for mobile cloud computing environments," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr./May 2014, pp. 404–409.

[16] J. Li, K. Bu, X. Liu, and B. Xiao, "ENDA: Embracing network inconsistency for dynamic application offloading in mobile cloud computing," in *Proc. 2nd ACM Mobile Cloud Comput. (MCC) Workshop*, Hong Kong, 2013, pp. 39–44.

[17] W. Gao, "Opportunistic peer-to-peer mobile cloud computing at the tactical edge," in *Proc. IEEE Military Commun. Conf.*, Baltimore, MD, USA, Oct. 2014, pp. 1614–1620.

[18] W. Liu, R. Shinkuma, and T. Takahashi, "Opportunistic resource sharing in mobile cloud computing: The single-copy case," in *Proc. 16th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Hsinchu, Taiwan, Sep. 2014, pp. 1–6.

[19] A. W. Min, R. Wang, J. Tsai, M.-A. Ergin, and T.-Y. C. Tai, "Improving energy efficiency for mobile platforms by exploiting low-power sleep states," in *Proc. 9th Conf. Comput. Frontiers*, Cagliari, Italy, May 2012, pp. 133–142.

[20] F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang, and J. Ma, "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing," *Inf. Syst. Frontiers*, vol. 16, no. 1, pp. 95–111, Mar. 2014.

[21] 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 13th Int. Conf. Grid Comput.*, Beijing, China, Sep. 2012, pp. 139–146.

[22] W. Zhang, Y. Wen, and D. O. Wu, "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 190–194.

[23] P. Shu, F. Liu, H. Jin, M. Chen, F. Wen, and Y. Qu, "eTime: Energy-efficient transmission between cloud and mobile devices," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 195–199.

[24] S. Chen, Y. Wang, and M. Pedram, "Optimal offloading control for a mobile device based on a realistic battery model and semi-Markov decision process," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2014, pp. 369–375.

[25] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conf. HotCloud*, 2010, p. 4.

[26] S. A. Saab, A. Chehab, and A. Kayssi, "Energy efficiency in mobile cloud computing: Total offloading selectively works. Does selective offloading totally work?" in *Proc. 4th Annu. Int. Conf. Energy Aware Comput. Syst. Appl. (ICEAC)*, Istanbul, Turkey, Dec. 2013, pp. 164–168.

[27] G. Han, A. Qian, J. Jiang, N. Sun, and L. Liu, "A grid-based joint routing and charging algorithm for industrial wireless rechargeable sensor networks," *Comput. Netw.*, vol. 101, pp. 19–28, Jun. 2016.

[28] G. Han, L. Wan, L. Shu, and N. Feng, "Two novel DOA estimation approaches for real-time assistant calibration systems in future vehicle industrial," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–12, Jun. 2015.

[29] C.-X. Wang *et al.*, "Cellular architecture and key technologies for 5G wireless communication networks," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 122–130, Feb. 2014.

[30] X. Ma, Y. Cui, and I. Stojmenovic, "Energy efficiency on location based applications in mobile cloud computing: A survey," in *Proc. 9th Int. Conf. Mobile Web Inf. Syst. (MobiWIS)*, vol. 10. 2012, pp. 577–584.

[31] M. S. Bali and S. Khurana, "Effect of latency on network and end user domains in cloud computing," in *Proc. Int. Conf. Green Comput., Commun. Conservation Energy (ICGCE)*, Chennai, India, Dec. 2013, pp. 777–782.

[32] Z. Wan, "Sub-millisecond level latency sensitive cloud computing infrastructure," in *Proc. Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Moscow, Russia, Oct. 2010, pp. 1194–1197.

[33] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Commun. Surveys Tut.*, vol. 18, no. 3, pp. 2220–2243, 3rd Quart., 2016.

[34] M. Zamith *et al.*, "A distributed architecture for mobile digital games based on cloud computing," in *Proc. Brazilian Symp. Games Digit. Entertainment (SBGAMES)*, Salvador, Brazil, Nov. 2011, pp. 79–88.

[35] Y. Cui, X. Ma, H. Wang, I. Stojmenovic, and J. Liu, "A survey of energy efficient wireless transmission and modeling in mobile cloud computing," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 148–155, Feb. 2013.

[36] S. S. Saini, R. Bagga, D. Singh, and T. Jangwal, "Architecture of mobile application, security issues and services involved in mobile cloud computing environment," *Int. J. Comput. Electron. Res.*, vol. 1, no. 2, pp. 58–67, Aug. 2012.

[37] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proc. 1st Int. Workshop Mobile Cloud Comput. Netw.*, Bengaluru, India, Jul. 2013, pp. 19–26.

[38] P. Si, Q. Zhang, F. R. Yu, and Y. Zhang, "QoS-aware dynamic resource management in heterogeneous mobile cloud computing networks," *China Commun.*, vol. 11, no. 5, pp. 144–159, May 2014.

[39] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration," *IEEE Commun. Surveys Tut.*, vol. 15, no. 1, pp. 362–386, 1st Quart., 2013.

[40] C. Peoples, G. Parr, and S. McClean, "Context-aware characterisation of energy consumption in data centres," in *Proc. 12th IFIP/IEEE Workshop Manage. Future Internet*, Dublin, Republic of Ireland, May 2011, pp. 1250–1257.

[41] G. W. Musumba and H. O. Nyongesa, "Context awareness in mobile computing: A review," *Int. J. Mach. Learn. Appl.*, vol. 2, no. 1, pp. 1–10, May 2013.

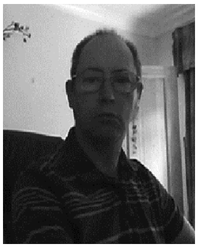
[42] X. Zhang, B. Li, and J. Zhu, "A combinatorial auction-based collaborative cloud services platform," *Tsinghua Sci. Technol.*, vol. 20, no. 1, pp. 50–60, Feb. 2015.

[43] S. Ji and B. Li, "Wide area analytics for geographically distributed datacenters," *Tsinghua Sci. Technol.*, vol. 21, no. 2, pp. 125–135, Apr. 2016.

[44] J. Donovan. (2011). *Bluetooth Goes Ultra-Low-Power*. [Online]. Available: http://www.digikey.com/en/articles/techzone/2011/dec/bluetooth-goes-ultra-low-power

[45] A. Communications. (2016). *Network Technologies*. [Online]. Available: http://www.axis.com/global/en/learning/web-articles/technical-guide-to-network-video/network-technologies
[46] Spec. (Feb. 17, 2016). *First Quarter 2016 SPECpower_ssj2008 Results*. [Online]. Available: https://www.spec.org/power_ssj2008/results/
[47] Y. Grunenberger. (2014). *On 4G, 3G, 2G Power Consumption of a E3276 Stick*. [Online]. Available: https://grunenberger.net/2014/02/24/4g-3g-2g-power-consumption/
[48] I. Wallossek and C. Angelini. (2015). *Skylake: Intel's Core i7-6700K and i5-6600K*. [Online]. Available: http://www.tomshardware.co.uk/skylake-intel-core-i7-6700k-core-i5-6600k,review-33276-11.html
[49] B. Dorman. (2015). *TomTom MyDrive Brings Satnav Syncing to PCs and Mobiles*. [Online]. Available: http://www.theregister.co.uk/2015/04/30/tomtom_mydrive_launch/

JOHN PANNEERSELVAM (M'14) is currently pursuing the Ph.D. degree with the University of Derby, U.K. He is currently a Graduate Teaching Assistant with the University of Derby. He has authored notable peer-reviewed international conferences, journals, and as book chapters. His current research is focused on energy efficient cloud systems. His research interests include cloud computing, big data, AI techniques, opportunistic networking, and P2P computing.



JAMES HARDY is currently pursuing the Ph.D. degree with the School of Computing and Mathematics, The University of Derby. Besides manufacturer specific data networking qualifications. He has authored or co-authored papers on traffic control, IoP, virtualisation, IPv6 addressing, software aging, and web services location. He has industrial experience including aerospace control systems and global scale wide area data networking. His research interests include smart city transportation, autonomous vehicles, communication systems, green computing, HaaS, IaaS, control systems, simulation, and virtualization.

LU LIU (M'07) received the Ph.D. degree from the University of Surrey. He is currently a Professor of Distributed Computing with the School of Computing and Mathematics, University of Derby, and an Adjunct Professor with the School of Computer Science and Communication Engineering, Jiangsu University. His research interests are in areas of cloud computing, social computing, service-oriented computing, and P2P computing. He is a fellow of British Computer Society.

BO YUAN received the B.Sc. degree in computer science and technology from Tongji University, Shanghai, China, in 2011. He is currently pursuing the joint Ph.D. degree with Tongji University and the University of Derby. His research interests include service discovery, online social networks, mobile computing, and peer-to-peer systems.

NICK ANTONOPOULOS received the Ph.D. degree in computer science from the University of Surrey in 2000. He is currently the Dean of the College of Engineering and Technology with the University of Derby, and the University of Derby Technical Coordinator of the framework collaboration with CERN and the ALICE experiment. He has over 18 years of academic experience. He has authored over 150 articles in fully refereed journals and international conferences. His research interests include cloud computing, P2P computing, software agent architectures, and security.