# Energy-Efficient Computation Offloading for Multimedia Workflows in Mobile Cloud Computing

Tao Huang[1,2], Yi Chen[3], Shengjun Xue[1,3], Haojun Ji[2], Yuan Xue[3], Lianyong Qi[4(✉)], and Xiaolong Xu[3]

[1] School of Computer Science and Technology, Silicon Lake College, Suzhou, China
nuisthuangtao@163.com
[2] Shanghai Jiading District Meteorological Bureau, Jiading, China
[3] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[4] School of Information Science and Engineering, Qufu Normal University, Qufu, China
lianyongqi@gmail.com

**Abstract.** In recent years, mobile cloud computing (MCC) is utilized to process multimedia workflows due to the limitation of battery capacity of mobile devices, which influences the experience of multimedia applications on the mobile devices. Computation offloading based on cloudlet is introduced as a novel paradigm to relieve the high latency which offloading computation to remote cloud causes. However, it is still a challenge for mobile devices to offload computation of multimedia workflows in cloudlet-based cloud computation environment to reduce energy consumption, which meets time constraints at the same time. In view of the challenge, an energy-efficient computation offloading method of multimedia workflow with multi-objective optimization is proposed in this paper. Technically, an offloading method based on cloudlet using Differential Evolution (DE) algorithm is proposed to optimize the energy consumption of the mobile devices with time constraints. Finally, massive experimental evaluations and comparison analysis validate the efficiency of our proposed method.

**Keywords:** Energy-efficient · Offloading · Multimedia workflow · Mobile · Cloudlet · DE

## 1 Introduction

The unparalleled improvements in mobile devices have changed the way that people enjoy multimedia services [1]. Audio and video documents are accessed and played easily from mobile devices. However, resource intensive multimedia applications like video encoding are in need of data processing and transmitting. Consequently, the multimedia applications drain mobile devices battery rapidly and cause high consumption of battery energy [1, 2].


H. Gao et al. (Eds.): TridentCom 2018, LNICST 270, pp. 113–123, 2019.
https://doi.org/10.1007/978-3-030-12971-2_7

Mobile cloud computing (MCC) could enhance the capacities of mobile devices for resource intensive multimedia applications through offloading computation to the resource-rich cloud [3, 4]. Benefit from MCC, video encoding which is extraordinarily heavy task for a mobile phone without efficient encoding applications should be offloaded to the cloud to save much energy. However, the long distance between users and clouds could cause unpredictable delay. For the sake of reducing the latency, cloudlet is proposed as an alternative to remote clouds. Cloudlets are server clusters which are co-located with wireless Access Points (APs), and mobile users can reduce the delay by offloading latency-intensive and resource-intensive applications to local cloudlets. Comparing with the local MCC, cloudlets provide powerful computing resource closer to enhance the efficiency of cloud system [5, 6].

Whereas in reality, owing to the distributed multimedia production environment, organized multimedia workflows are desperate for maximum quality [7, 8]. For offloading multimedia workflows to remote cloud cause the high latency, offloading computation in cloudlet-based cloud computing environment is introduced to relieve the latency. With the observations above, it is still a challenge for mobile devices to offload computation of multimedia workflows in cloudlet-based cloud computation environment with a multi-objective optimization. In view of this challenge, an offloading method for multimedia workflow using Differential Evolution (DE) algorithm is proposed.

In this paper, we have made the following contributions. Firstly, we model multimedia applications by workflows, and every operation in multimedia workflows is modeled as a series of tasks in the multimedia workflow. And we analyze the average waiting time of the tasks in the cloudlet, the processing and transmission time of tasks. Then we propose an energy-efficient computation offloading method for multimedia workflow through DE algorithm to optimize the model's fitness function and constraints. Finally, the best individual is obtained through a series of mutation, crossover and selection as the solution output.

## 2 Problem Formulation

In this chapter, the basic concepts are proposed. Modeling multimedia applications by workflow, Optimizing multimedia workflows based on "Cloudlet", Taking the energy consumption of the mobile devices as the evaluation target. According to the priority of the multimedia workflows, the time consumption is used as a constraint.

### 2.1 Problem Modeling

This chapter will model the multimedia applications by workflows. Every multimedia application is modeled as a multimedia workflow, every operation in multimedia applications will be modeled as a series of tasks in the multimedia workflows, such as intelligent analysis/audit, presetting parameter template, transcoding, information extraction, content repair and encryption, etc. And the ordinal relationship between tasks will be represented by the directed graph, for example, the directed graph $G_m(V_m, \beta_m)$ represents the ordinal relationship of each task and the amount of data transferred between each task in the m-th multimedia workflow, $V_m = \{v_{1,m}, v_{2,m}, \ldots, v_{N,m}\}$ represents the set of tasks in the m-th multimedia workflow, N is the number of tasks in the m-th multimedia workflow, $\beta_m = \{r(v_{i,m}, v_{j,m}), d_{i,j} | i \neq j, 1 \leq i, j \leq N\}$ is the set of dependencies between tasks, $r(v_{i,m}, v_{j,m})$ represents $v_{i,m}$ is the predecessor task of $v_{j,m}$, $v_{j,m}$ only be executed after $v_{i,m}$ is completed, there may also be a situation where a task has more prerequisite tasks, and $d_{i,j}$ is the amount of data transferred between $v_{i,m}$ and $v_{j,m}$.

The following figure shows an example of multimedia workflow represented by the directed graph for transcoding, extracting, and encrypting video. And this multimedia workflow is mainly to transcode, extract information and encrypt the video uploaded by the mobile devices. But in reality, the different multimedia workflows may contain different operations tasks, and the execution relationships or sequence of tasks may be different (Fig. 1).

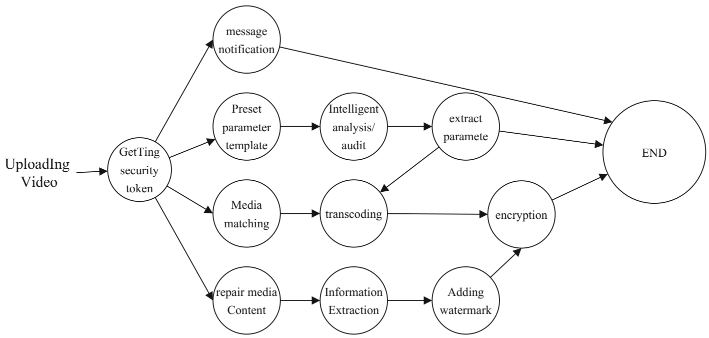



**Fig. 1.** An example of multimedia workflow for transcoding, extracting, and encrypting video

Usually, the tasks of multimedia workflows are executed on the mobile devices or remote cloud. However, due to the limitations of the hardware performance of the mobile devices and the high latency of the remote cloud. For the multimedia workflows that require high hardware performance and high real-time, the mobile devices and the remote cloud are not so ideal because of their own limitations. Cloudlet is widely used in the processing of multimedia workflows because it features "the higher performance than mobile devices and the lower latency than remote cloud".

Cloudlet is configured as multiple Virtual Machines (VMs) for concurrent processing the multimedia workflow, which is modeled as a 3-tuple, denoted as $CL = (N_{VM}; P_{cl}; D_{LAN})$, where $N_{VM}$ is the number of VMs in the cloudlet, $P_{cl}$ is the power of the cloudlet and $D_{LAN}$ is the transmission delay in Local Area Network (LAN).

Every task in the multimedia workflows is modeled as a 2-tuple $v_{i,m} = (w_{i,m}, s_{i,m})$, where $w_{i,m}$ is the workload of task $v_{i,m,}$ and $s_{i,m}$ is the offloading strategy of task $v_{i,m}$. Because the DE algorithm uses real coding, the offloading strategy is expressed as a one-dimensional real vector $S = \{s_{i,m} | i = 1, 2, \ldots, N_m; m = 1, 2, \ldots, M\}$, where $-0.5 \leq s_{i,m} < 0.5$ represents that the task $v_{i,m}$ is performed locally, $-1.5\, d_{i,j}\, s_{i,m} < -0.5$

represents that $v_{i,m}$ is offloaded to the cloudlet and $0.5 \leq s_{i,m} < 1.5$ represents that $v_{i,m}$ is offloaded to the cloud, respectively. $N_m$ represents the number of the tasks in the m-th multimedia workflow and $M$ represents the number of the multimedia workflows.

## 2.2 Analysis of Multimedia Workflow

In this section, the model of multimedia workflows will be analyzed mainly, including: the average waiting time of the tasks in the cloudlet, the processing and transmission time of tasks, energy and time computation model, and the object function.

**Average Waiting Time Analysis.** Unlike the remote cloud services that have almost unlimited resources and the mobile devices that only serve the users themselves, the resource-constrained cloudlet usually need to serve multiple users around it. When the number of tasks in the task queue exceeds the number of virtual machines in the cloudlet, the subsequent tasks need to wait.

Assuming that the number of virtual machines in a cloudlet is $N_{VM}$, the arrival rate of tasks in the cloudlet obeys the Poisson distribution whose mean is $\lambda$, and the service time of each task in the cloudlet obeys the Exponential Distribution whose mean is $\mu$. If the task queue in the cloudlet system is infinite length, and the tasks in the queue obey the "First In First Served" principle. According to the Queuing Theory, the "*M/M/M/* $\infty$" model is used.

**Processing and Transmission Time Analysis.** If the network bandwidth is $BW$, the transmission time between tasks $v_{i,m}$ and $v_{j,m}$ can be calculated as follows:

$$T_{trans}(v_{i,m}, v_{j,m}) = \frac{d_{i,j}}{BW} \tag{1}$$

The task execution time mainly includes: Waiting Time $T_{wait}$, Calculation Time $w_{i,m}/p$ and Network Delay $D$. If the offloading strategy of the task is different, the composition and calculation methods of execution time are also different. The waiting time $T_{wait}$ is only happen when the task is executed in the cloudlet. The calculation time $w_{i,m}/p$ is related to the efficiency of the execution device, and the network delay $D$ is related to the data transmission network (LAN or WAN). The calculation model of specific execution time is as (2). And $p_l$, $p_{cl}$, $p_c$ represent the power of the mobile devices locally, cloudlet and cloud respectively, $D_{LAN}$, $D_{WAN}$ are the transmission delay in Local Area Network (LAN) and Wide Area Network (WAN).

$$T_{exe}(v_{i,\mathrm{m}}) = \begin{cases} \frac{w_{i,m}}{p_l}, & -0.5 \leq s_{i,m} < 0.5 \\ T_{wait} + \frac{w_{i,m}}{p_{cl}} + D_{LAN}, & -1.5 \leq s_{i,m} < -0.5 \\ \frac{w_{i,m}}{p_c} + D_{WAN}, & 0.5 \leq s_{i,m} < 1.5 \end{cases} \tag{2}$$

### 2.3 Energy and Time Computation Model

The energy consumption of data transmission between $v_{i,m}$ and $v_{j,m}$ is:

$$E_{trans}(v_{i,m}, v_{j,m}) = \frac{d_{i,j}}{BW} * p_{\mathrm{t}} \tag{3}$$

$p_t$ is the power of mobile devices during data transmission;

When the task is executed on the mobile devices, the mobile devices are active at this time and the power $p$ is $p_a$. On the contrary, when the task is executed in the cloudlet or the cloud, the mobile devices are idle and the power $p$ is $p_i$. So the power consumption of $v_{i,m}$ is

$$E_{\mathrm{exe}}(v_{i,m}) = T_{\mathrm{exe}}(v_{i,m}) * p \tag{4}$$

Then, the total energy consumption of the mobile devices is calculated as follows:

$$E_{app,m}(S) = \sum_{v_{i,m} \in V} E_{exe}(v_{i,m}) + \sum_{r(v_{i,m}, v_{j,m}) \subseteq \beta} E_{trans}(v_{i,m}, v_{j,m}), \quad m \in \{1, 2, \ldots, M\}. \tag{5}$$

### 2.4 Object Function

The total energy consumption of the mobile devices is used as the evaluation target, and the deadline of each multimedia workflow is used as a constraint. Due to each multimedia workflow has a different priority. The scaling factor $F_{PR}$ is used to scale or expand the execution time of the multimedia workflows.

## 3 An Energy-Efficient Computation Offloading Method for Multimedia Workflow Using DE Algorithm

This chapter mainly encodes the energy-efficient computation offloading model of multimedia workflows, and optimizes the model's fitness function and constraint using the DE Algorithm, the non-dominated sorting approach and the crowded-comparison are used in the optimization process. Through a series operations of mutation, crossover and selection, the best individual is finally obtained as the optimal solution output. Below we will give a detailed optimization process:

**Algorithm 1.** Calculate the completion time for the multimedia workflow

**Input:** m-th multimedia workflow, offloading strategy $S$
**Output:** Completion time of the multimedia workflow $T_{app,m}(S)$
$pre_v=\emptyset, D^*=\emptyset, T_{app,m}(S)=0$
**if** $v_{j,m}$ is the first node in $G_m(V, \beta)$ **then**
    $D(v_{j,m})=T_{exe}(v_{j,m})$
**else**
    add $v_{j,m}$ to $pre_v$
**end if**
**for** $v_{j,m} \in pre_v$ do
    $D(v_{j,m})= D(v_{i,m})+T_{trans}(v_{i,m}, v_{j,m})+T_{exe}(v_{j,m})$
    add $D(v_{j,m})$ to $D^*$
**end for**
**for** $d \in D^*$ do
    **if** $T_{app,m}(S)<d$ **then**
        $T_{app,m}(S)=d$
    **end if**
**end for**
**return** $T_{app,m}(S)$

### 3.1 Encoding

DE algorithm using real-coded, so each task of multimedia workflows has an integer index organized from 0 and a real value representing an offloading strategy of the task. The offloading strategies of all tasks in the multimedia workflows are combined into an individual. Each individual represents an implementation of multimedia workflows and a population consists of multiple individuals. The DE algorithm performs evolutionary operations on each generation of population to seek the optimal execution strategy for the optimization problem.

### 3.2 Fitness Function and Constraint

In this paper, the DE algorithm uses the energy consumption of the mobile devices as the fitness function to find the offloading strategies that can make the total energy consumption the lowest. Each deadline of multimedia workflows is a constraint, so that each multimedia workflow must be completed within the deadline, Algorithm 1 explains how to calculate the completion time of multimedia workflow. Then DE algorithm performs evolutionary operations on every generation of population to find the optimal execution strategy for multimedia workflows.

### 3.3 Optimize the Multimedia Workflows Using DE Algorithm

**Initialization.** The size of individual $N$ is determined by the number of tasks in all multimedia workflows. Each value of gene in the individual represents the offloading strategies of the tasks corresponding to the index. The offloading strategy values range from $[-1.5, 1.5)$, and its meaning has already been introduced in the Basic Concepts section. Then the experiment needs to determine the parameters of the DE algorithm,

such as the appropriate number of evolutionary iterations $G$ and the population size $2NP$. Finally, the experiment also needs to determine the appropriate crossover Probability $CR$, the mutation probability $F$ and mutation strategy.
The first-generation populations $P_1$ is generated by randomly assigning values for each gene of each individual in the range [−1.5, 1.5).

**Evolution.** Combining the multiple evaluation target, The "Pareto sorting" is used for the population to obtain a set $L$ with multiple layers of dominating solutions, and calculate the "crowd distance" for each layer of dominating solutions. From the first layer of dominating solution, the preferred individuals are selected to fill in $R_{g+1}$ until the population size of $R_{g+1}$ is $NP$. Then DE algorithms performs evolutionary operations on the population $R_{g+1}$, including mutations, crossovers, and selections to generate a new populations $O_{g+1}$ whose size is also $NP$. Finally, $R_{g+1}$ and $O_{g+1}$ are combined into the next-generation populations $P_{g+1}$, and the optimization continues. Until the termination condition is satisfied, the optimal offloading strategies and the energy consumption for all mobile devices are output.

## 4 Comparison and Analysis of Experimental Results

This chapter mainly uses four kinds of offloading methods: Benchmark, CBO, CLBO and DECO. Among them, Benchmark, CBO and CLBO represent that all tasks of multimedia workflows will be executed in mobile devices, remote cloud and cloudlet respectively. DECO assigns the tasks of multimedia workflows to the mobile devices, remote cloud, and cloudlet for execution using the DE algorithm intelligently, to optimize the energy consumption of the mobile devices.

### 4.1 Experimental Settings

In this experiment, the performance of the method was tested by optimizing different numbers of multimedia workflows. Each multimedia workflow contains 13 tasks, and the workload for tasks in every multimedia workflow are set to {0; 20; 50; 8; 20; 30; 80; 60; 30; 50; 20; 40; 0}. But the ordinal relationship of each task and the amount of data transferred between each task in the different multimedia workflow are different. In this experiment, the parameter settings are as follows: the power of mobile devices when the CPU is idle and active are 0.001 W and 0.5 W, respectively, and the transmission power of mobile devices is 0.1 W. The processing capacity of mobile devices, cloudlet and cloud are 500 MHz, 2000 MHz and 3000 MHz, respectively. The delay of LAN and WAN are 1 ms and 30 ms, respectively. The LANs and WANs bandwidth are 100 kb/s and 50 kb/s. The average waiting time of tasks in the cloudlet is 20 ms. And the experimental datasets can be accessed in the following link: https://drive.google.com/open?id=1nc-QVdthUCjUSzazDgYxMIh0EGEN-r_L.

## 4.2 Performance Evaluation

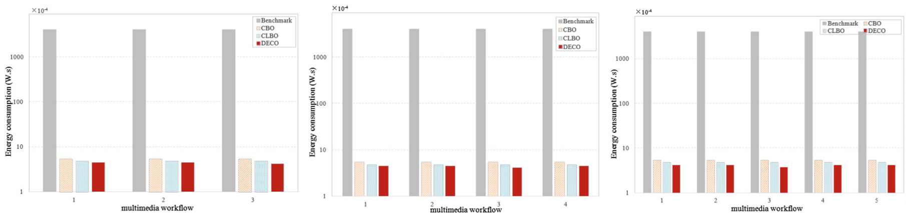
In this section, three groups of experiments were conducted by selecting 3, 4 and 5 multimedia workflows: Figs. 2 and 3 show the comparison of energy consumption and completion time of mobile devices and Fig. 4 shows the assignment of all tasks for multimedia workflows using four kinds of offloading methods in three groups of experiments.

It can be clearly seen from Figs. 2 and 3 that in the three groups of experiments, energy consumption and completion time using the Benchmark offloading method are much higher than the other three methods. Because the computing performance of the mobile devices is so limited compared to remote cloud and cloudlet that it will certainly consume a lot of energy and computing time. However, when the tasks of multimedia workflows are offloaded to the remote cloud or the cloudlet, the mobile devices only needs to transmit data and wait for results instead of performing calculation tasks that consume a lot of energy and time.

Secondly, Figs. 2 and 3 also show that CLBO is better than CBO because CLBO has lower network latency than CBO.

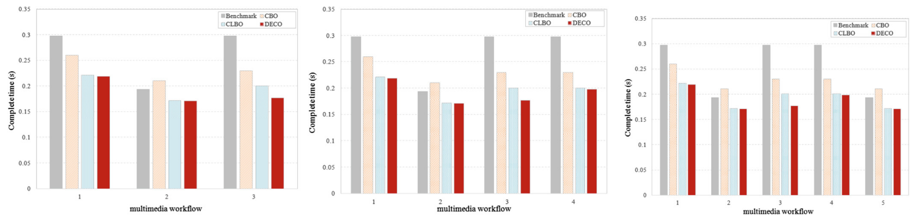
But this does not mean that the performance of CLBO will always be better than CBO, because CBO has higher computational performance than CLBO. and when there are too many tasks in the cloudlet, the tasks will have waiting time. They have their own advantages and disadvantages. However, there is no doubt that the effect of DECO must be better than the other three methods. This can also be clearly seen from the experimental results. Because during the optimization of multimedia workflows using the DE algorithm, the other three offloading methods are just the special states of DECO. For example, the strategy of CLBO and CBO is to take the value of each gene of every individual in the population between [−1.5, −0.5) and [0.5, 1.5) respectively.

The three groups of experimental results in Fig. 4 show that in this experiment, the effect of DECO was better than the other three offloading methods. It is precisely because of the optimization of multimedia workflows using the DE algorithm that the tasks of multimedia workflows are reasonably allocated to the corresponding devices (mobile devices, cloudlet, and remote cloud).



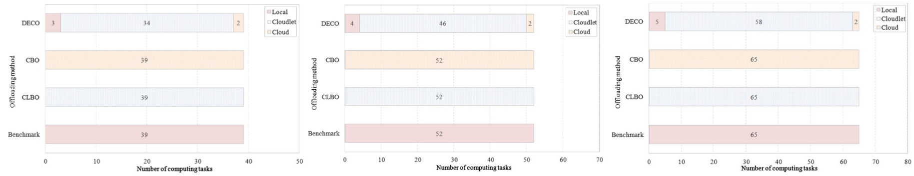

(a) Number of Workflows = 3 (b) Number of Workflows = 4 (c) Number of Workflows = 5

**Fig. 2.** The comparison of energy consumption




(a) Number of Workflows = 3 (b) Number of Workflows = 4 (c) Number of Workflows = 5

**Fig. 3.** The completion time




(a) Number of tasks = 39 (b) Number of tasks = 52 (c) Number of tasks = 65

**Fig. 4.** The assignment of all tasks for multimedia workflow

At the same time, through the results of this group of experiments, it can be seen that the reason that the effect of DECO is closer to CLBO is that most of the tasks in multimedia workflows are still assigned to execute in the cloudlet.

Finally, it can be confirmed that the DECO is definitely the best one among the 4 offloading methods. And this has been verified in this experiment. However, due to the limitation of the scale of this experimental data set, the respective advantages of the remote cloud and the cloud cannot be fully reflected. In practical applications, the scale and complexity of multimedia workflows running in remote cloud and cloudlet are far greater than the data sets of this experiment. At that time, the optimization effect of multimedia workflows using DECO will be even more obvious.

## 5 Related Work

In recent years, with the development of multimedia applications on mobile devices, MCC which has faster data processing rate is beneficial to process data from multimedia applications [9–12]. Additionally, offloading multimedia workflows which have complex functionality to the cloudlets to save energy and processing time of multimedia applications [13, 14].

Deng et al. proposed and optimized a computation computing strategy for workflows which are related to mobile devices. Liu et al. proposed a hybrid model which transferring data through multiple wireless networks, and formulated the offloading problem in hybrid wireless networks in [10]. In [14], Shah-Mansouri et al. developed a dynamic task scheduler which could decide the offloading strategy on the cloud service provider side (CSP) and determined the optimal pricing strategy via profit maximization. He et al. constructed a new privacy-aware authentication (PAA) scheme which has less computation costs for MCC services [11]. In [13], Elgendy et al. proposed a novel framework which utilizes an optimization model to decide the offloading strategy based on energy consumption, CPU use, implementing time, and memory utilization to offload intensive computation tasks to the cloud from the mobile device.

In [10], Zhang et al. modeled the waiting time of the cloudlet and proposed a hybrid computing offloading algorithm which minimizes the all energy consumption that mobile devices cause. Besides, Rimal et al. presented a novel cloudlet-aware resource management scheme to reduce the offload delay [12].

To the best of our knowledge, there are few researches on the multimedia workflows scheduling through offloading multimedia workflows to the cloudlets with multi-objective optimization. With the observations above, it is still a challenge to realize an efficient multimedia workflow offloading method which reduces the energy consumption and meets the time constraint. In view of this challenge, an offloading method for multimedia workflow using DE algorithm is proposed.

## 6 Conclusion and Future Work

The existing multimedia workflow offloading methods barely take both time and energy consumption into account. In this paper, an energy-efficient computation offloading method of multimedia workflow based on cloudlet is proposed. We analyze execution time and construct a model of energy and time computation of mobile device. The method based on cloudlet using DE algorithm is designed to optimize the energy consumption of the mobile devices within deadline. Through ample experimental evaluation, validate the efficiency of our proposed method is proved.

Based on the work done in this paper, we will adjust and extend our multimedia workflow offloading method to reduce the energy and time consumption of multimedia applications. We will implement the offloading method to improve the experience of users when uploading videos, and keep updating our method according to the real-life performance.

**Acknowledgement.** This research is supported by the Research Project of Shanghai Meteorological Bureau Scientific under Grant No. TD201807 and the National Science Foundation of China under grant no. 61702277, no. 61672276, no. 61772283, no. 61402167 and no. 61672290, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120, and Natural Science Foundation of Jiangsu Province (Grant No. BK20171458). Besides, this work is also supported by The Startup Foundation for Introducing Talent of NUIST, the open project from State Key Laboratory for Novel Software Technology.

## References

1. Kaewmahingsa, K., Bhattarakosol, P.: Mobile cloud system: a solution for multimedia retrieval via mobile phones. In: International Conference on Computing & Convergence Technology, vol. 8652, no. 5, pp. 36–40 (2012)
2. Altamimi, M., Palit, R., Naik, K., Nayak, A.: Energy-as-a-Service (EaaS): on the efficacy of multimedia cloud computing to save smartphone energy. In: IEEE Fifth International Conference on Cloud Computing, pp. 764–771 (2012)
3. Chen, X.: Decentralized computation offloading game for mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. **26**(4), 974–983 (2015)
4. Kovachev, D., Yu, T., Klamma, R.: Adaptive computation offloading from mobile devices into the cloud. In: IEEE International Symposium on Parallel & Distributed Processing with Applications, pp. 784–791 (2012)
5. Liu, Y., Lee, M.: Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. IEEE Trans. Mob. Comput. **15**(10), 2398–2410 (2016)
6. Xu, Z., Liang, W., Xu, W., et al.: Efficient algorithms for capacitated cloudlet placements. IEEE Trans. Parallel Distrib. Syst. **27**(10), 2866–2880 (2016)
7. Hazekamp, N., Kremer-Herman, N., Tovar, B., Meng, H., Choudhury, O.: Combining static and dynamic storage management for data intensive scientific workflows. IEEE Trans. Parallel Distrib. Syst. **29**(2), 338–350 (2018)
8. Liu, P., Wang, R., Ding, J., Yin, X.: Performance modeling and evaluating workflow of ITS: real-time positioning and route planning (1), 1–15 (2017)
9. Deng, S., Huang, L., Taheri, J., Zomaya, A.Y.: Computation offloading for service workflow in mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. **26**(12), 3317–3329 (2015)
10. Zhang, J., et al.: Hybrid Computation offloading for smart home automation in mobile cloud computing. Pers. Ubiquit. Comput. **22**(1), 121–134 (2018)
11. He, D., Kumar, N., Khan, M.K., et al.: Efficient privacy-aware authentication scheme for mobile cloud computing services. IEEE Syst. J. **12**(2), 1621–1631 (2018)
12. Li, R., Shen, C., He, H., Xu, Z., Xu, C.Z.: A lightweight secure data sharing scheme for mobile cloud computing. IEEE Trans. Cloud Comput. **6**(2), 344–357 (2018)
13. Elgendy, I., Zhang, W., Liu, C., Hsu, C.: An efficient and secured framework for mobile cloud computing. IEEE Trans. Cloud Comput. (2018)
14. Xue, S., Peng, Y., Xu, X., Zhang, J., Shen, C., Ruan, F.: DSM: a dynamic scheduling method for concurrent workflows in cloud environment. Cluster Comput. **3**, 1–14 (2017)