A Balanced Cloudlet Management Method for Wireless Metropolitan Area Networks

Xiaolong Xu[1,2,3](✉), Yuhao Chen[1,2], Lianyong Qi[4], Jing He[1,2],
and Xuyun Zhang[5]

[1] School of Computer and Software,
Nanjing University of Information Science and Technology, Nanjing, China
`njuxlxu@gmail.com`
[2] Jiangsu Engineering Centre of Network Monitoring,
Nanjing University of Information Science and Technology, Nanjing, China
[3] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
[4] School of Information Science and Engineering, Qufu Normal University,
Jining, China
[5] Department of Electrical and Computer Engineering, University of Auckland,
Auckland, New Zealand

Abstract. With the rapid development of wireless communication technology, cloudlet-based wireless metropolitan area network, which provides people with more convenient network services, has become an effective paradigm to meet the growing demand for requirements of wireless cloud computing. Currently, the energy consumption of cloudlets can be reduced by migrating tasks, but how to jointly optimize the time consumption and energy consumption in the process of migrations is still a significant problem. In this paper, a balanced cloudlet management method, named BCM, is proposed to address the above challenge. Technically, the Simple Additive Weighting (SAW) and Multiple Criteria Decision Making (MCDM) techniques are applied to optimize virtual machine scheduling strategy. Finally, simulation results demonstrate the effectiveness of our proposed method.

Keywords: Cloudlet · WMAN · VM migration ·
Energy consumption · Time consumption

1 Introduction

In recent years, to meet the growing demand for the requirements of wireless cloud computing, wireless metropolitan area network (WMAN) has emerged as a public network in the society, providing people with more convenient network services [1]. Computing tasks of mobile devices are migrated from mobile devices to remote data centers through access points in WMAN. With the increasing

H. Gao et al. (Eds.): TridentCom 2018, LNICST 270, pp. 62–75, 2019.
https://doi.org/10.1007/978-3-030-12971-2_4

popularity of mobile devices and the maturing of wireless communication technology, WMAN achieves great development with its excellent features in coverage, transmission speed and comprehensive cost [2].

Nowadays, the wide spread of smart phones and their powerful functions have made them an important part of many people's life over the world [3]. A variety of mobile applications have been developed to bring people more convenience and enrich their lives. However, the content-rich and powerful applications increase the resource cost of mobile devices. Due to many restrictions, such as large quantities of energy consumption and application running time, mobile devices are often difficult to meet the resource requirements of the applications [4].

As a new type of business computing mode, cloud computing can provide users with high reliability, dynamic and scalable virtual computing resources in a on-demand way. Users can enter the configurable computing resources shared pool and get access to resources quickly [5].When faced with complex mobile applications which require a large amount of computation source, mobile devices often choose to migrate these tasks to a cloud resource center for better performance and less energy consumption [6]. In a complex wireless network environment, for the traditional mobile cloud computing architecture, a cloud is generally located far away from mobile users, which inevitably leads to the communication delay between mobile devices and the cloud [7]. In service framework of WMAN, cloudlets as edge devices are widely used to reduce end-to-end communication delay [8,9].

Cloudlets are deployed closed to users to provide rich data storage and computing resources [10]. Generally, a large number of cloudlets and access points are deployed in WMAN environment. Users can easily access to cloudlets through APs in WMAN for fast and efficient computation [8]. With the constant promotion and popularity of cloud computing technology, the number of cloudlet consumers is increasing rapidly. When there are only few tasks executing on a cloudlet, the running cloudlet will waste large quantities of energy. Compared with it, it will be more energy-saving to migrate the tasks to other cloudlets and close the running cloudlet. Generally, users migrate computing tasks to a cloudlet in the form of virtual machine (VM) [11]. Though the number of running cloudlets decreases sharply after several migrations, excessive time consumption and energy consumption during migrations are both big problems which are often ignored.

In the view of this challenge, a VM scheduling method for balancing energy consumption and time consumption in wireless metropolitan area network is proposed in this paper. The main contributions of this paper can be summarized in three folds. Firstly, we formulate a problem for reducing the two objectives, time consumption and energy consumption. Secondly, a balanced cloudlet management method, named BCM, is proposed to achieve the goal. In this method, the Simple Additive Weighting (SAW) and Multiple Criteria Decision Making (MCDM) techniques are applied to optimize scheduling strategy. Finally, comprehensive experiments are conducted to verify the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 formulates a problem for reducing energy consumption and time consumption in cloudlet environment. In section 3, a balanced cloudlet management method named BCM is proposed. Then several experiments are conducted, and the results are presented in Sect. 4. Finally, Sect. 5 analyzes the related work and Sect. 6 states the conclusion and future work.

2 Problem Formulation

In this section, We formulate a problem for reducing energy consumption and time consumption in WMAN. Key terms used in the formulation are listed in Table 1.

Table 1. Key terms and descriptions

Term	Description
C	The set of running cloudlets, $C = \{c_1, c_2, \ldots, c_N\}$
W	The set of access point(AP), $W = \{w_1, w_2, \ldots, w_Q\}$
V	The set of running VM, $V = \{v_1, v_2, \ldots, v_M\}$
$E_{Active}(t)$	The energy consumption of the running VMs
$t^n_{MP}(t)$	The running time of the n-th cloudlet
$E_{Idle}(t)$	The energy consumed by idle VMs
$E_{Base}(t)$	The basic energy consumption of cloudlets
$T^n_{edge}(t)$	The time consumed by transmission between cloudlets and APs
$T^n_{mid}(t)$	The time consumed by migration from a AP_{sour} to the AP_{desc}
$P_q(t)$	The energy consumption rate of the q-th AP W_q
$E_{AP}(t)$	The energy consumption of APs
$E_{Oper}(t)$	The energy consumption of opening/closing operation
$E_{All}(t)$	All the energy consumption
$T_{All}(t)$	All the time consumption

2.1 Basic Concepts

We assume that there are N cloudlets and Q access points in WMAN, denoted as $C = \{c_1, c_2, \ldots, c_N\}$ and $W = \{w_1, w_2, \ldots, w_Q\}$, respectively. Consider a scenario, only one physical machine(PM) is deployed in each cloudlet. Besides, we assume that there are M VMs running in cloudlets, denoted as $V = \{v_1, v_2, \ldots, v_M\}$.

Figure 1. shows an example of task migration in cloudlet-based WMAN. There are a certain amount of APs between two cloudlets. The AP next to the source cloudlet and the AP next to the destination cloudlet are named as AP_{sour} and AP_{desc}, respectively. The AP between a AP_{sour} and a AP_{desc} is named as AP_{mid}.

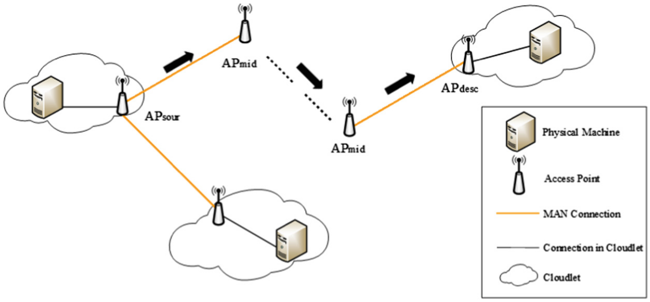


Fig. 1. An example of task migration in cloudlet-based WMAN

2.2 Time Consumption Model of VM Migration

To reduce the energy consumption, we shall migrate the VMs from hosted cloudlets to other cloudlets. However, the process of migration consumes a quantities of transmission time.

$I_m^n(t)$ is a binary variable to judge whether the v_m is placed on cloudlet c_n at time t, which is defined by

$$I_m^n(t) = \begin{cases} 1, & \text{if } \mathrm{v}_m \text{ is placed on } \mathrm{c}_n \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$M_m^{n,n'} t$ is a binary variable that indicates whether v_m is migrated from c_n to $c_{n'}$ at time t, which is defined as

$$M_m^{n,n'}(t) = \begin{cases} 1, & \text{if } \mathrm{v}_m \text{ is migrated from } \mathrm{c}_n \text{ to } \mathrm{c}_{n'} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

When the cloudlet c_n needs to be transferred between the cloudlet and the AP next to it, the time consumption is calculated by

$$T_{edge}^n(t) = \sum_{m=1}^{M} \sum_{n'=1}^{N} I_m^n(t) \cdot M_m^{n,n'}(t) \cdot \frac{d_m}{\rho}, \tag{3}$$

where d_m is the data size of the v_m, and ρ is the transmission efficiency between the cloudlet and the AP next to it.

Let $G_{n,n'}$ be the number of APs between the c_n and the $c_{n'}$, including a AP_{sour}, a AP_{desc} and several AP_{mid}s. The time consumed by migration from AP_{sour} to AP_{desc} is calculated by

$$T_{mid}^n(t) = \sum_{m=1}^{M} \sum_{n'=1}^{N} I_m^n(t) \cdot M_m^{n,n'}(t) \cdot \frac{d_m}{s} \cdot (G_{n,n'} - 1), \tag{4}$$

where s is the transmission efficiency between APs.

During the entire process of migration, the time consumption is calculated by

$$T_{All}(t) = \sum_{n=1}^{N} (2 \cdot T_{edge}^{n}(t) + T_{mid}^{n}(t)) \tag{5}$$

2.3 Energy Consumption Model of VM Migration

Based on [12], the energy consumption of the running VMs is calculated by

$$E_{\text{Active}}(t) = \sum_{m=1}^{M} \alpha_m \cdot \varphi_m(t) \tag{6}$$

where α_{m} is the energy consumption rate of the m-th VM, and $\varphi_{\mathrm{m}}(t)$ indicates the execution time of the m-th VM v_m.

A cloudlet will continue running until all the tasks in this cloudlet have been excuted, thus the running time of the n-th cloudlet is calculated by

$$t_{MP}^{n}(t) = \max_{m=1}^{M} \{ I_m^n(t) \cdot \varphi_m(t) \} \tag{7}$$

The energy consumed by idle VMs is calculated by

$$E_{Idle}(t) = \sum_{n=1}^{N} \sum_{m=1}^{M} \beta_m \cdot (t_{MP}^{n} - \varphi_m(t)) \tag{8}$$

where β_m is the energy consumption rate of the m-th VM v_m in idle mode.

The basic energy consumption of cloudlets is calculated by

$$E_{Base}(t) = \sum_{n=1}^{N} \gamma_n \cdot t_{MP}^{n}(t) \tag{9}$$

where γ_n is the basic energy consumption rate of the n-th cloudlet.

During the process of migration, APs shall consume energy due to the data transmission. Based on [13], the energy consumption rate of the q-th AP w_q is calculated by

$$P_q(t) = e_q + \eta_q p_q \cdot \sum_{p \in W} \frac{h_q}{r_{pq}} \tag{10}$$

where e_q, h_q, p_q, η_q and r_{pq} is the baseline power of the q-th AP w_q, the traffic demand of w_q, the signal transmission power of w_q, the signal transceiver power factor of w_q and the link rate between w_p and w_q, respectively.

The energy consumption of APs is calculated by

$$E_{AP}(t) = \sum_{q=1}^{Q} \int_{\mathrm{t}}^{T_{All}(t)} P_q(t) dt \tag{11}$$

In this paper, the operation time of VM starting and closing is set to the same constant, which is denoted as g. The energy consumption of the starting/closing operation of the VM is calculated by

$$E_{Oper}(t) = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{n'=1}^{N} 2g \cdot I_m^n(t) \cdot M_m^{n,n'}(t) \cdot \alpha_m \tag{12}$$

Based on the energy consumption model in [14], let $E_{All}(t)$ be all the energy consumed by both the APs and cloudlets, which is calculated by

$$E_{All}(t) = E_{Active}(t) + E_{Idle}(t) + E_{Base}(t) + E_{Oper}(t) + E_{AP}(t) \tag{13}$$

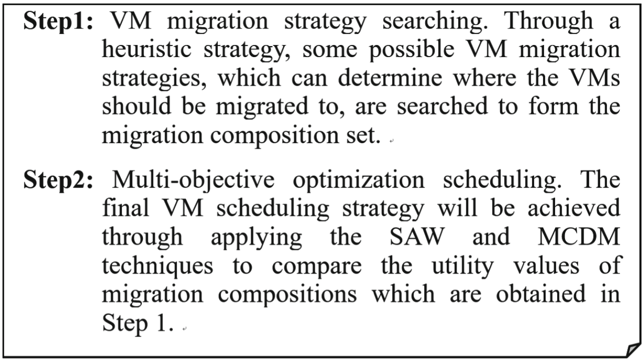
2.4 Problem Definition

From the above analysis, energy consumption model and time consumption model are built to quantify the two objectives. In this paper, how to evaluate the two influencing factors synthetically and minimize the consumption is our core issue. The problem is formulated by

$$min\ T_{All}(t),\ min\ E_{All}(t) \tag{14}$$

3 Balanced Cloudlet Management Method for WMAN

Based on the analysis in Sect. 2, a balanced cloudlet management method is proposed in this section. Our method mainly consists of two steps as in Fig. 2.



Step1: VM migration strategy searching. Through a heuristic strategy, some possible VM migration strategies, which can determine where the VMs should be migrated to, are searched to form the migration composition set.

Step2: Multi-objective optimization scheduling. The final VM scheduling strategy will be achieved through applying the SAW and MCDM techniques to compare the utility values of migration compositions which are obtained in Step 1.

Fig. 2. Specification of balanced cloudlet management method.

3.1 VM Migration Strategy Searching

The tasks hosted on a cloudlet usually have different numbers of VM requirements. Here, we assume that each cloudlet has only one PM. In Algorithm 1, we propose a method to find VM migration strategy.

Definition 1 (migration strategy ms_k): We define the scheme migrating a VM from hosted cloudlet to destination one as a migration strategy, which is denoted as mc_k=(sp_k, ep_k, tc_k), where sp_k, ep_k and tc_k are the hosted cloudlet of the k-th VM, the destination cloudlet of the k-th VM and the time consumption of migration, respectively.

In this method, we only migrate VMs to the cloudlets with idle space, and the cloudlets which are running in full load will not be migration targets.

Definition 2 (migration composition mc_k): Through analysis, all the VMs of the same task should be migrated to the same cloudlet, thus we may get more than one migration strategy recording the source selection and allocation in the VM scheduling process at the same time. Generally, there are several tasks in one cloudlet. We name the set of migration strategies which migrate all the tasks of one cloudlet as a migration composition.

In Step 1, we sort the cloudlets in the decreasing order of idle VMs firstly, and migrate tasks from their hosted cloudlets to the destination cloudlets which have as little but enough idle space as possible. In the migration process, there may be more than one cloudlet to meet the requirements. To deal with these scene, we calculate the time consumption values caused by different choices with the formula (5) and select the strategy with minimum time consumption value. Then a temporal migration strategy is generated. If all the VMs in this cloudlet can be migrated away, a migration composition will be generated. Otherwise, the temporal strategies will be deleted. After acquiring a migration composition, the idle cloudlet will be turn off and all these cloudlets will be resorted. The next migration composition is generated by iteration. In other words, new migrations will be created on the basis of previous one until all the VMs can't be migrated.

Figure 3 shows an example of migration destination selection. c_1, c_2, c_3 and c_4 are four cloudlets running in a wireless metropolitan area network. Two tasks t_1 and t_2 need to be executed on c_1 which occupy two VMs and one VM respectively. Both of c_2 and c_3 meet the requirements of migration of t_1. However, the migration from c_1 to c_2 costs 1.1s, while the migration from c_1 to c_3 only costs 1s. In this condition, we will select the second strategy as ms_1. Task t_2 only occupies one VM which c_4 can provide adequately. Thus we select c_4 as the migration target of t_2. After the two migration, the cloudlet c_1 is idle, and the set $\{ms_1, ms_2\}$ is denoted as mc_1.

We assume there are N cloudlets, i.e., $\{rs_1, rs_2, \ldots, rs_N\}$. A standby cloudlet set SPM is used to store the cloudlets which have idle space. Once a migration composition is generated, the full cloudlets shall be removed from SPM, and SPM shall be updated. We assume there are K migration compositions generated, i.e., $\{mc_1, mc_2, \ldots, mc_K\}$. Because the next migration composition is generated by iteration, the next migration composition need cover all the migration strategies in previous composition.

In Algorithm 1, we firstly sort the cloudlets in the decreasing order of idle VMs and create the standby cloudlet set which stores the cloudlets with idle space (Lines 1 to 7). Then we analyze each task, select the appropriate cloudlet to host all the VMs of the task and generate migration compositions (Lines 8 to 28).

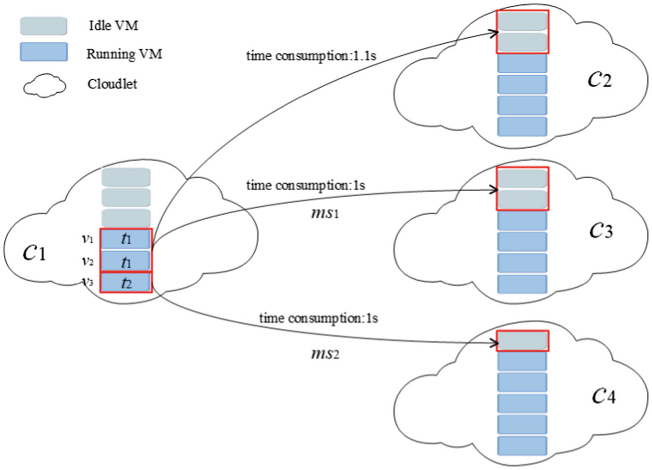


Fig. 3. An example of migration target selection

3.2 Multi-objective Optimization Scheduling

Algorithm 1 generates all the migration compositions which meet the requirements. Then in this section, the Simple Additive Weighting (SAW) and Multiple Criteria Decision Making (MCDM) techniques [12] are applied to multi-objective resource scheduling.

The SAW method is mainly used to calculate the weight of several criteria. The QoS criteria are divided into two categories, positive criterion and negative criterion. In this experiment, energy consumption and time consumption are both negative criteria whose quality decreases as its value increase. Algorithm 1 is designed to acquire the migration composition set MC. Each migration composition in MC brings about an energy consumption value and a time consumption value. Let $E_{All} = ({E_{All}}^i(t), 1 \leq i \leq K)$ be the energy consumption value and $T_{All} = ({T_{All}}^i(t), 1 \leq i \leq K)$ be the time consumption value. According to the specifications above, the utility value generated by migration is calculated by:

$$U_i(t) = \frac{E_{All}^{\max}(t) - E_{All}^{i}(t)}{E_{All}^{\max}(t) - E_{All}^{\min}(t)} \cdot w_E + \frac{T_{All}^{\max}(t) - T_{All}^{i}(t)}{T_{All}^{\max}(t) - T_{All}^{\min}(t)} \cdot w_T \tag{15}$$

In this formula, $E_{All}^{\max}$ and $E_{All}^{\min}$ represent the maximum and minimum energy consumption in all migration composition respectively. $T_{All}^{\max}$ and $T_{All}^{\min}$ represent the maximum and minimum time consumption in all composition respectively. w_E and w_T are the weight of energy consumption and the weight of time consumption.

With the formula above, the utility value of each migration composition can be calculated. The MCDM technique is mainly used to make the final determination in migration compositions. We select the composition with the maximum utility value as the final scheduling strategy. In addition, in order to save the

Algorithm 1. VM Migration Strategy Searching

```
Require: The running cloudlet set $RS = \{rs_1, rs_2, \ldots, rs_N\}$
Ensure: The migration composition set $MC = \{mc_1, mc_2, \ldots, mc_K\}$
Sort the cloudlets in cloudlets in the decreasing order of idle VMs,$PM = \{pm_1, pm_2, \ldots, pm_N\}$
for $n$=1 to $N$ do
    Get the number of idle VMs in $PM_n$,i.e.,$c_n$
    if $c_n > 0$ then
        Add $pm_n$ to the standby cloudlet set,i.e.,$SP_M$
    end if
end for
$flag = 0, n = 1$
while $flag = 0 \&\& n \leq N$ do
    Get the VM list on the $n$-th cloudlet in $SPM$
    Get the task list on the $n$-th cloudlet in $SPM$
    for each task do
        Get the number of occupied VMs
        Select the cloudlets with least idle space to host the VMs
        Calculate $T_All$ for all selected cloudlets
    end for
    Generate a temporal strategy with smallest $T_{All}$
    if all the VMs can be migrated away then
        Generate a migration composition $mc_n$
        Add $mc_n$ to $MC$
        Update the $SPM$
    else
        $flag = 1$
        Delete the temporal strategy
    end if
    $n = n + 1$
end while
return $MC$
```

energy consumption, we need to set the vacant cloudlets to the sleeping mode after finishing the migration.

In Algorithm 2, we firstly calculate energy consumption value and time consumption value of each migration composition (Lines 2 to 5). From them, we get the maximum and minimum value of both energy and time (Lines 6 to 7). Then we use the formula (15) to calculate the utility values of each migration composition (Lines 8 to 10). Finally, we select the best scheduling strategy and set all the vacant cloudlets to the sleeping mode (Lines 11 to 12).

4 Experiment Evaluation

In this section, a serious of comprehensive experiments are conducted to evaluate performance of our proposed balanced cloudlet management method for wireless

Algorithm 2. Multi-objective Optimization Scheduling

```
Require: The migration composition set $MC$
Ensure: VM scheduling strategy
for each migration composition in $MC$ do
    Calculate the energy consumption value $E_{All}$
    Calculate the time consumption value $T_{All}$
end for
Get maximum and minimum energy consumption values
Get maximum and minimum time consumption values
for each migration composition in $MC$ do
    Calculate the utility values
end for
Schedule the VMs by the strategy with maximum utility value
Set the vacant cloudlets to the sleeping mode
```

metropolitan area network. In the process of comparison, the running state without migration is marked as a benchmark, our proposed method is abbreviated as BCM.

4.1 Experimental Context

In our experiment, HP ProLiant ML 110 G4 is selected as the cloudlet server to create the cloud infrastructure services network. Its basic configuration consists Intel Xeon 3040, Dual-Processor clocked at 1860 MHz and 4 GB of RAM. The baseline power of server, the baseline power of VM and the running power of VM are 86 W, 6 W and 4 W, respectively. Seven basic parameters and the range of values in this experiment are shown in Table 2. More specifically, 6 different-scale datasets are generated for our experiment, and the number of running cloudlets are 50, 100, 150, 200, 250 and 300, respectively.

4.2 Performance Evaluation

In this section, we analyze the energy consumption, resource utilization efficiency and transmission time consumption to validate our proposed method.

Energy Consumption Evaluation. In the experiment, the energy consumption here consists three parts, VM energy consumption, cloudlet energy consumption and transmission energy consumption. According to the experiment, about half of the cloudlets will be turn off after migrations. With the decrease of the number of running cloudlets, the cloudlet energy consumption will decrease, while the transmission energy consumption will increase. Figure 4 illustrates that BCM method saves 17.7%, 22.2%, 23.1%, 23.3%, 23.7% and 25.3% of energy when the number of cloudlets is 50, 100, 150, 200, 250 and 300, respectively. It can be concluded that the energy consumption in the BCM method is much less than the energy consumption in the benchmark case. To summary, our proposed BCM method can achieve the goal of saving energy.

Table 2. Parameter settings

Parameter	Domain
Number of running cloudlets	{50, 100, 150, 200, 250, 300}
Number of running VMs in each cloudlet	[1, 6]
Number of VMs on each cloudlet	6
Transmission rate between APs (Mb/s)	540
Transmission rate between AP and cloudlet (Mb/s)	1200
VM duration time	[1, 3]
VM transmission data (Gb)	[0.5, 0.8]

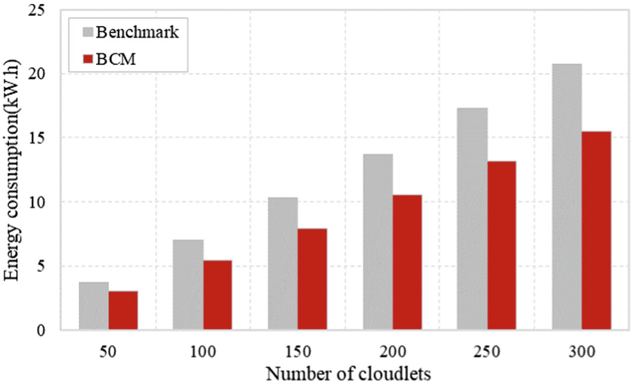


Fig. 4. Comparison of total energy consumption with Benchmark and BCM.

Resource Utilization Efficiency. Resource utilization efficiency is a dimension to calculate the average proportion of resources used in all cloudlets. As is shown in Fig. 5, the result illustrates that BCM method improves 28.3%, 40.0%, 35.3%, 33.7%, 39.7% and 43.1% of resource utilization efficiency when the number of cloudlets is 50, 100, 150, 200, 250 and 300, respectively. Stated thus, our proposed BCM method has much better resource utilization efficiency than benchmark case.

Transmission Delay Time. VM migrations between APs result in the time consumption, which has a negative impact on the performance. With the increase of the number of APs which VMs need to be migrated through, the time consumption will have a sharp growth. From Fig. 6, simulation results show that the average migration time consumption fluctuates around 3.8 s in the simulation environment.

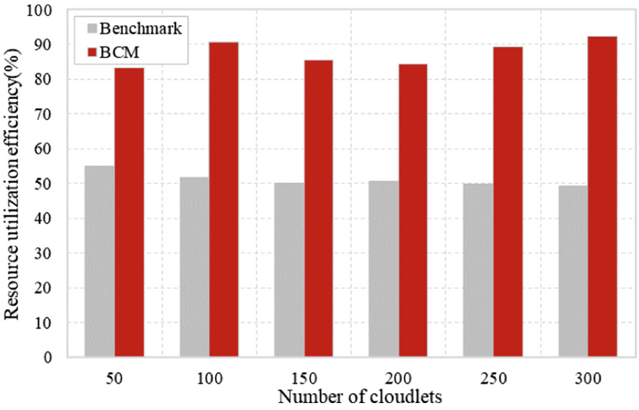


Fig. 5. Comparison of resource utilization efficiency with Benchmark and BCM.

5 Related Work

Recently, the growing market requirements of wireless broadband promotes the development of WMAN. Many advantages, such as fast speed and economical cost, make it popular among users. Nowadays, more and more complex applications are used in mobile devices, such as virtual reality, augmented reality, and interactive games. Due to the limitations of mobile device including battery, computing power, and CPU resources, users in urban cities intend to accomplish these tasks in cloud environment increasingly. However, the tremendous transmission delay is one significant limitation of migrating tasks to the remote clouds which is far away from mobile users [2,15]. In order to reduce the delay, cloudlets as a new technology, are deployed in WMAN [5]. Users can migrate computing-intensive tasks to the local cloudlets to achieve better user experience.

With the large-scale use of cloudlets, when the cloudlets are in the state of low load, the problems of energy consumption are often ignored. In other words, a cloudlet will waste a large quantities of energy if there are only few tasks executing on it. Therefore, it is necessary for us to find a optimal migration scheduling strategy.

To reduce the expected response time of migrating tasks to cloudlets, Li et al. [16] put forward a series of cloudlet server deployment issues in WMAN and proposed two novel cloudlet server deployment strategies, which achieve the goal of saving time by reducing the expected response time of the system. Liu et al. [17] proposed a cloudlet selection model based on mixed integer linear programming (MILP) and a resource allocation model based on MILP, in order to enable better performance in terms of access latency and resource utilization. Mudassar Ali et al. [18] proposed a delay minimization optimization problem under maximum workload and delay constraints. They formulated an optimization problem for joint cloudlet selection and latency minimization in fog network, and solved it with a distributed and self-organizing solution.

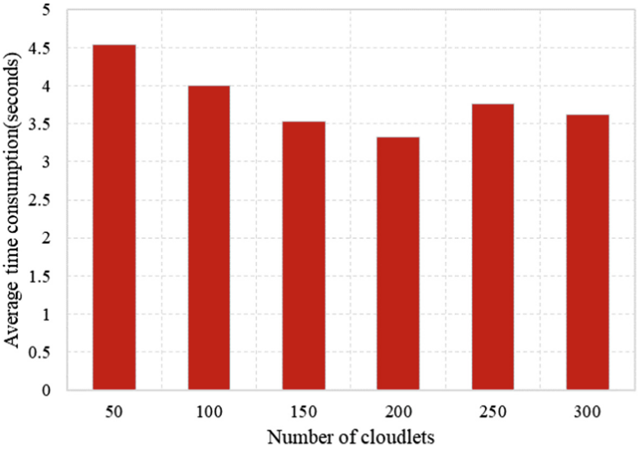


Fig. 6. Time consumption with BCM.

From the above analysis, it can be concluded that few investigations and studies have considered the joint optimization for both energy consumption and time consumption in cloudlet-based WMAN.

6 Conclusion and Future Work

In this paper, we formulated a problem for reducing energy consumption and time consumption in wireless metropolitan area network. In order to solve the challenge, a balanced cloudlet management method named BCM is proposed. In our method, the SAW and MCDM techniques are applied to optimize scheduling strategy. Finally, we carry out experiments to verify our proposed method.

For future work, we plan to apply our proposed method to real-world cloudlet-based WMAN and carry further investigation.

References

1. Baskaran, S.B.M., Raja, G.: Blind key distribution mechanism to secure wireless metropolitan area network. CSI Trans. ICT **4**(2–4), 1–7 (2016)
2. Yuan, C., Li, X., Wu, Q.M.J., Li, J., Sun, X.: Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. Comput. Mater. Contin. **53**(4), 357–371 (2015)
3. Lo'ai, A.T., Bakheder, W., Song, H.: A mobile cloud computing model using the cloudlet scheme for big data applications. In: 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), pp. 73–77. IEEE (2016)
4. Jin, A.-L., Song, W., Zhuang, W.: Auction-based resource allocation for sharing cloudlets in mobile cloud computing. IEEE Trans. Emerg. Top. Comput. **6**(1), 45–57 (2018)

5. Pang, Z., Sun, L., Wang, Z., Tian, E., Yang, S.: A survey of cloudlet based mobile computing. In: International Conference on Cloud Computing and Big Data, pp. 268–275 (2016)
6. Zhang, Y., Niyato, D., Wang, P.: Offloading in mobile cloudlet systems with intermittent connectivity. IEEE Trans. Mob. Comput. **14**(12), 2516–2529 (2015)
7. Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Netw. **24**(5), 2795–2808 (2016)
8. Shen, J., Tan, H.-W., Wang, J., Wang, J.-W., Lee, S.-Y.: A novel routing protocol providing good transmission reliability in underwater sensor networks. 網際網路技術學刊 **16**(1), 171–178 (2015)
9. Pan, Z., Zhang, Y., Kwong, S.: Efficient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Trans. Broadcast. **61**(2), 166–176 (2015)
10. Xiang, H., et al.: An adaptive cloudlet placement method for mobile applications over GPS big data. In: Global Communications Conference, pp. 1–6 (2017)
11. Dolui, K., Datta, S.K.: Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. In: Global Internet of Things Summit (GIoTS), pp. 1–6. IEEE (2017)
12. Xu, X., Zhang, X., Khan, M., Dou, W., Xue, S., Yu, S.: A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. Futur. Gener. Comput. Syst. (2017, in Press)
13. Garroppo, R.G., Nencioni, G., Procissi, G., Tavanti, L.: The impact of the access point power model on the energy-efficient management of infrastructured wireless lans. Comput. Netw. **94**, 99–111 (2016)
14. Xu, X., Dou, W., Zhang, X., Chen, J.: Enreal: An energy-aware resource allocation method for scientific workflow executions in cloud environment. IEEE Trans. Cloud Comput. **4**(2), 166–179 (2016)
15. Kaur, J., Kaur, K.: A fuzzy approach for an IoT-based automated employee performance appraisal. Comput. Mater. Contin. **53**(1), 24–38 (2015)
16. Li, D., Wu, J., Chang, W.: Efficient cloudlet deployment: local cooperation and regional proxy. In: 2018 International Conference on Computing, Networking and Communications (ICNC), pp. 757–761. IEEE (2018)
17. Liu, L., Fan, Q.: Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment. IEEE Access **6**, 24533–24542 (2018)
18. Ali, M., Riaz, N., Ashraf, M.I., Qaisar, S., Naeem, M.: Joint cloudlet selection and latency minimization in fog networks. IEEE Trans. Ind. Inform. **PP**(99), 1–8 (2018)