「マルチメディア，分散，協調とモバイル
(DICOMO2014) シンポジウム」 平成26年7月

A Just-in-Time Task Allocation Method and its Simulation Result in Wearable-Mobile Computing Environment

Zhitao Deng[†1] Peng Li[†2] Zixue Cheng[†2]

Abstract: Mobile Cloud Computing (MCC) is a hot research topic which tries to effectively use the resources for maximizing the throughput. However, MCC has not considered the experience of a user. And now wearable devices are becoming the next potential terminals, the time between two actions of using a wearable device is crucial for usability. Wearable-Mobile Cloud Computing (WMCC) is to improve user's just-in-time experience. In this paper, we aim to improve the experience in WMCC Environment. We propose an effective method for task allocation which optimizes the waiting time of the user between two consecutive user actions, and perform simulation to show the performance of the solution.

***Keyword*:** Ubiquitous mobile computing, wearable devices, resource allocation, just-in-time scheduling, genetic algorithms.

1. Background

Nowadays, the cloud computing has a great achievement. The cloud computing provides an easy way to help people solve the problem of the data storage, data process, and the data sharing. And with the development of the mobile device, base on the cloud computing, some researchers proposed a new type computing which is called Mobile Cloud Computing.

Mobile Cloud Computing (MCC), integrates the mobile computing and cloud computing [1]. Thanks to the pervasiveness of mobile devices and cloud computing, the MMC brings the great accessibility for users everywhere. However the traditional MMC considers the computing throughput, but pay less attention to the user experience.

With the development of wearable devices (WD), the wearable device has great potential to be the next influential product in the near future. Comparing with the mobile device, wearable device is more advanced in improving user experience. Therefore, our group wants to introduce a new type of computing which is called Wearable Mobile Cloud Computing. The Wearable Mobile Cloud Computing connects the wearable device to the mobile network, which enables the accessibility to the resource in cloud. But for each user, the bandwidth is different, and the users request for the real-time computing and improvement of theirs experience. Users desire the experience of using the devices conveniently, e.g. quick response time and mobile flexibility.

2. Related Works

This paper is most related to the Mobile Cloud Computing, but we aim to Wearable-Mobile Cloud Computing. Regarding Mobile Cloud Computing, Lei Yang [1][2] et al., argue the Component-as-a-Service(CaaS), and consider the components can be allocated and run in the different mobile devices and Cloud to finish a big job. In this case, they have defined the problem and find a way to solve this problem. They propose a genetic algorithm to solve the problem.

But in the field of the Wearable-Mobile Cloud Computing, we have to extend this problem. In the CaaS, the Component can be allocated to any mobile device. And the aim is to minimize the maximum span or throughput. In the wearable computing, the wearable device is driven by the user's behavior, and during the computing, the device has to give a feedback to the user interactively. Moreover, the wearable device also has to collect the user information, when the program is running. So, in the Wearable-Mobile Could Computing, some task has to be run in the wearable device, and other can run in the any device, e.g. a mobile device or a cloud. Based on this consideration, Fanxin [3] has defined the problem and did the theoretical study on the problem.

3. Purpose and Issues

Our purpose is to improve the user experience in WMCC environment. The good way to improve the user experience is to provide a just-in-time service, which means to minimize the time between two continuous actions of the user. In order to design the just-in-time system, we need to solve the following issues.

First, the definition of the just-in-time should be given. The just-in time is to let the response time less than the user's reaction time, but it will too hard for the design, because the limitation of the wearable device resources and the transmission speed between the wearable device and other devices. So, if the response time is less than a threshold, which is acceptable for

†1 The graduate school of computer science and engineering, The University of Aizu.
,Aizuwakamatsu City, Fukushima, 965-8580, Japan
Email:m5172103@u-aizu.ac.jp
†2 The school of computer science and engineering, The University of Aizu., Aizuwakamatsu City, Fukushima, 965-8580, Japan Email:z-cheng@u-aizu.ac.jp

the user, we say it is just-in-time.

Second, solution of finding the best allocation of tasks between the mobile devices and the cloud is necessary. A program/job can be divided into several tasks. A task can be run with the input and output data. The data also can drive other task to run. The problem is how to allocate the tasks to mobile device and cloud, so that the just-in-time property is satisfied. Since the problem is an NP hard problem, previous works of our research group have developed a Genetic Algorithm for the allocation problem [3][4].

4. Solution

For example, in order to perform a talk, which has to be done in the wearable device, the time of waiting for all input data from previous talks should be small enough i.e. less than a threshold. In other word, when a WD needs to run the task, all input should be ready. We try to allocate the talks, which should not be run on wearable device, to the mobile devices and could properly. We try to find as more just-in-time running tasks as possible, when considering the allocation. Therefore, we have developed a GA based solution in [3].

Fig.1 shows the model of the system. LR denotes local resource, i.e. mobile devices, RR shows the remote resource, i.e. cloud.

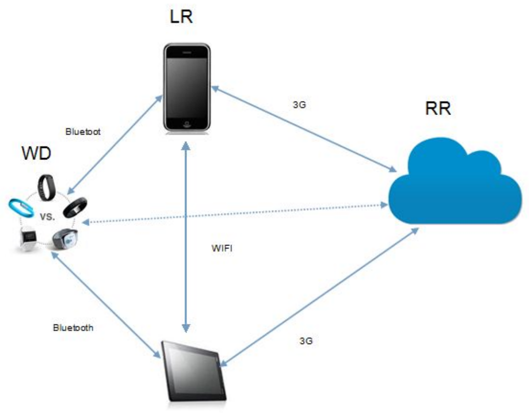

Fig. 1 The model of the System

Fig.2 shows the example of the program. The program/job is divided into tasks. We use a link to express the relationship between two tasks. The number with a link is the cost of the transmission, and the number within a circle (node) is the necessary running time of the task. The shadow circle means the tasks which have to be run in the wearable device, e.g. starting the program, termination of the program, and collecting the data of the user. This type of tasks can't be run in other devices but only in the wearable device. The white circle means that the task can be run in any device. Our objective is to allocate the tasks into cloud and other device (local device and wearable device) to optimize the system performance and improve the user experience.

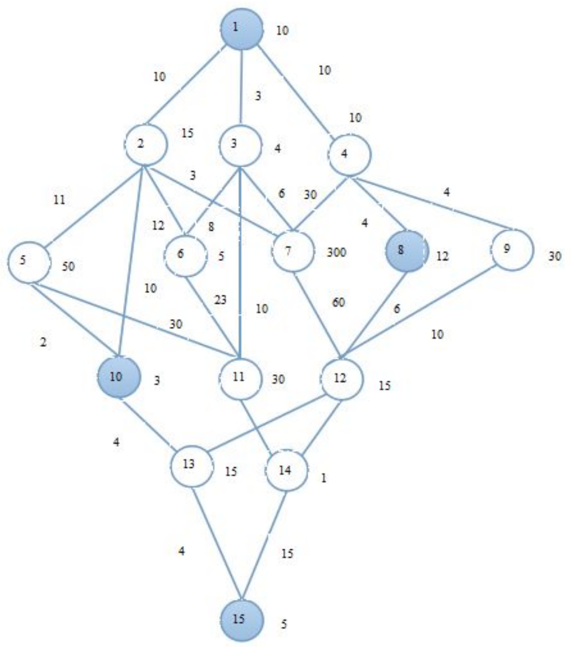

Fig. 2 The Task Graph

And we also aim to the just-in-time feature. We define the interval time between two continuous tasks, which have to be run in the wearable device as *waiting time (WT)*. With the introduction of the WT, we can know the just-in-time problem is to shorten WT problem. If the WT is too large, the user will feel boring about the response speed. For running a job/program, there are more than one WT, since there may be many interactions between the user and the wearable devices. It is very hard to guarantee each WT is small enough. We try to maximize the number of WTs which are acceptable. We assume a threshold to mean the acceptable upper limit of a user. And we try to maximize the number of WT such that WT is smaller than the threshold. Formally,

If $ET_i^{wd} - ET_{i-1}^{wd} \le \sigma$ then $z = z + 1$ end if

For each task in G:

$$\mathrm{T}_i^e = \max_{j \in prev(i)} \left\{ T_j^e + \frac{D_i}{C_i} + \sum_{j \in prev(i)} \frac{O_j}{S_j} \right\}$$

End for

In this paper, we will mainly focus on showing the simulation results. For the details of the definition and the GA algorithm for solving about problem, please refer to [3].

5. Simulation

5.1 Simulation setting

We have made performed some simulations, by changing the talk graph, the communication capacity between the devices, and the ratio of required communication amount to computation amount (RCC).

In order to make the simulation easily and understandably, we use several numbers as identifiers for different types of tasks. Specifically, we use 1 to demote WD limit task, i.e. this task has to be run in the wearable device; 2 to mean Non-WD task, i.e. this task can not be run in wearable device. We use WT to means the interval time between two consecutive tasks, which have to be run in the wearable device

(1) Suppose the cost assignment of tasks is according to the Gaussian distribution, we create the task graph in a random way. The tasks of the program/job will be different in different programs/jobs.

(2) Create the assignment of the transmission time by using the Gaussian distribution method. It will be not a real simulation if we use all the number for assignment of transmission time in random. Thus we use a rate to limit the assignment of the transmission. If the program is very large, the total assignment of the program will be in great variety. Therefore, we define the rate like that:

$$\varphi_{\mathrm{RCC}} = \frac{Assignment\_of\_task(T_1 + T_2 ... T_n)}{Assignment\_of\_transmission(T'_1 + T'_2 ... T'_m)}$$

(3) As we know the bandwidth of the network is uncertain. When we calculate the transmission time, we have to use the bandwidth, which is impacted by the network factor α . We have to get a new bandwidth when α changed.

Here is the pseudo-code of the GA program.

```
Input:
        Tasks
Output:
        Optimized allocation solution
Start Procedure：
        Run Top Level Finding Algorithm to find a order of tasks;
        Create the Chrome randomly;
    WHILE not achieve maximum loops (gen number)
        (1) Calculate the fitness of each Chrome;
        (2) Select the Chrome base on the fitness;
        (3) Recombinant the Chrome;
        (4) Check the number in the each Chrome, make sure the WDLimit area is the number of the wearable device.
        (5) Mutation New Chrome mutation
        (6) Check the number in the each new Chrome, make sure the WDLimit area is the number of the wearable device.

        (7) Calculate the fitness of new Chrome
        (8) Re-insert the offspring into the parent to get a new population.
        (9) The gen number plus 1;
    End WHILE
End Procedure
```

5.2 Simulation design

In order to perform the simulation for a verity of talks, we have developed a simulation program which can create a task graph randomly. Suppose the assignment of running time/cost of each task follows to the Gaussian distribution. The nodes (talks) and links (channels) are created in a random way. Tasks will be different in different programs.

The amount of computing and communication is also created based on the Gaussian distribution method. Since the computation massive tasks should be allocated to the cloud, and communication massive tasks should be allocated into local devices, the ratio RCC is crucial. We create computation amount and the transmission amount between two tasks, based on the ratio. Below we show three experiments, we have performed

Experiment 1: Verify the correctness of the program

We use the number 3 to represent of the *wearable device* and use the number 2 to represent the cloud, and use number 0 and number 1 to represent the local resource, such as an iPad or a Smartphone. And in the chrome we have identified some WD limit tasks, which are tasks 1, 8, 12, and 15. These tasks can be run only on the wearable device. Here we compare 3 different allocation methods. The position of the chrome represents the ID of the talk, and the number in the chrome means the device to which the task is allocated. For example, the first (most left item) is task 1. The most right item in the array means task

Case1: This is the trivial allocation, where all the tasks will be run in the wearable device, and the chrome design is as follow:

3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Case2: All non-WD tasks will be allocated to the cloud, and the chrome design is as follow:

3	2	2	2	2	2	2	3	2	2	2	3	2	2	3

Case3 (our method): Our method is used to allocate the tasks.

The chrome is as follow by running our simulation algorithm:

3	2	1	1	0	2	1	3	0	1	0	3	2	1	3

The above show our method can sure allocate the talks into different devices

Experiment 2: Evaluating the influence due to the number of tasks

We aim to optimize the just-in-time, so we have to get the total number of the tasks which satisfy that the WT is shorter than a given threshold. In the simulation, we will changing the number of tasks, and remain other parameters fixed. Because the task graph is created randomly, for the same number of the task, we may get a different result. This for each case, i.e., each number of tasks, we run the simulation 50 times and get the average value. Finally we will evaluate the proposal using the average value. In the experiment, we keep the WD limited task as 40% of the total talks.

Case1: Run the program by using graphs having 10 tasks.

Case2: Run the program by using graphs having 15 tasks

Case3: Run the program by using graphs having 20 tasks

Experiment 3: Evaluating the influence with RCC

We change the rate of the task and transmission. We can find the influence of the assignment of computing cost/time, and communication cost/time. We choose graph having 15 tasks to do this experiment, just by changing the RCC. We also use the same variable in the experiment 2.

Case1: Run the program by using the RCC equal to 0.35.

Case2: Run the program by using the RCC equal to 0.45.

Case3: Run the program by using the RCC equal to 0.55.

5.3 Simulation results

The simulation results of **Experiment1** is shown in **Fig. 3**

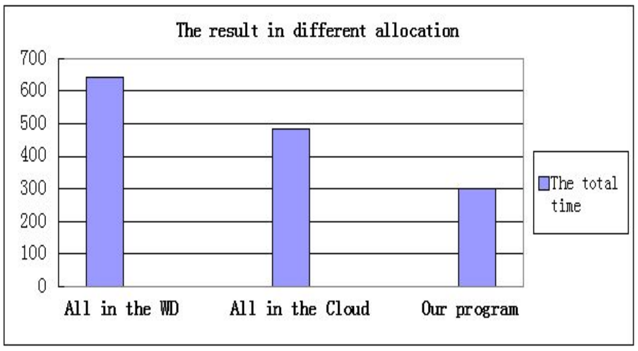

Fig. 3 Simulation results of Experiment1

Our program has shorted the running time among the 3 cases. That is to say simple way of allocation, such as allocating almost all tasks into the cloud is an easy way, but may not the best way.

In the Experiment 2, we want to show the solution is good for the random task and it will have a good result of experiment.

The Experiment2 simulation results are shown in **Fig. 4**

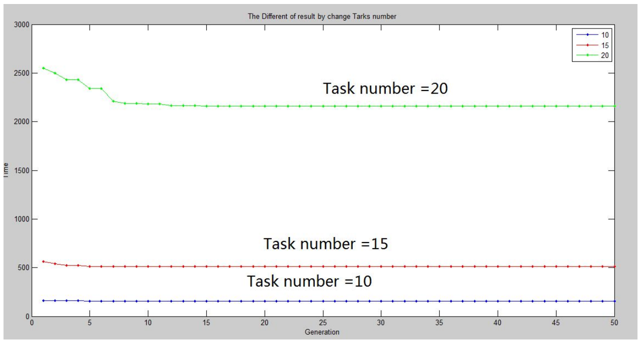

Fig. 4 Simulation results 1 of Experiment2

The x-axis in the result representing the generation of the GA algorithms, and y-axis means the total time. In this Figure, we can see the result becomes convergence. For different case, the generation No for convergence is different, but they all have good convergence, before 10 generations.

The compare results of **Experiment2** are shown in **Fig.5**

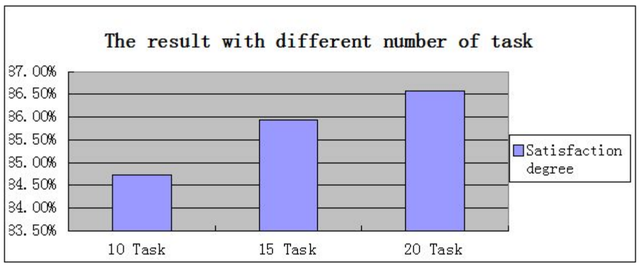

Fig. 5 Simulation results 2 of Experiment 2

The dark bar in histogram is satisfaction degree(the ratio of the average number of satisfied WT to total task number). In the result, 10 tasks's satisfaction degree is 84.74%, 15 tasks's satisfaction degree is 85.93%, 20 tasks's satisfaction degree is 86.58%. With increasing of number of task, satisfaction degree has increased from 84.74% to 86.58%.

The simulation results of **Experiment3** are shown in **Fig. 6**

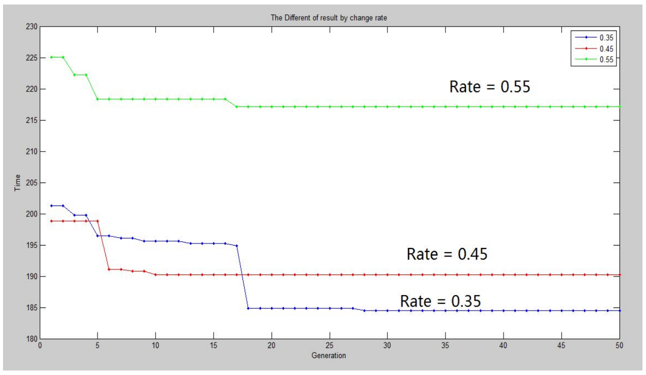

Fig. 6 Simulation result 1 of Experiment3

The x-axis represents the generation of the GA, and y-axis represents the total time. In this figure, we can see all the curve becomes convergence. The first curve is the ratio of 0.55, it means that the total cost is large than 0.45 and 0.35. The second curve is the ratio of 0.45, and the last curve is the ratio of 0.35.

The just-in-time task results are shown in **Fig.7**

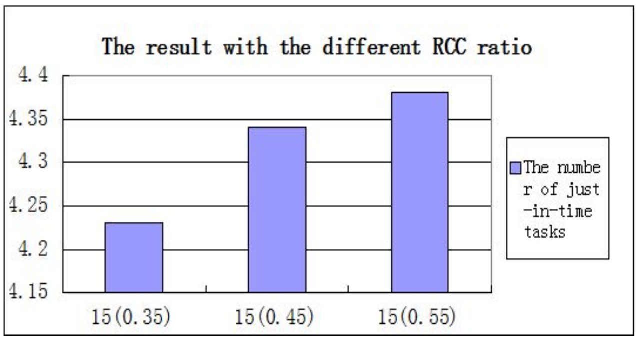

Fig. 7 Simulation result 2 of Experiment3

We can see if the rate is smaller, the satisfaction number(the average number of satisfied WT) is larger. For example, in the 15 tasks which is in the RCC ratio 0.35, we get the average number of satisfied WT is 4.23 from 100 times running result, and in the RCC ratio 0.45, we get the average number is the 4.34 from 100 times result. And in RCC ratio 0.55, we get the average number is the 4.38 from 100 times. The ratio is the 84.6%, 86.8%, 87.2%.

6. Conclusion

In this paper, we studied the task allocation problem for wearable device, mobile data stream application with the genetic algorithm. We conduct a series simulation to verify our method. The simulation results indicate that our method are effective on the task allocation problem. In our simulation, we have finished some experiments by using our simulation program. But we have not considered the transmission speed's influence to the result. Our future work is to consider the transmission speed change in the model, and do more experiment to show the effectiveness of the solution.

7. Acknowledgment

We would like to thank Mr. Fanxin for his master research work. This paper shows the additional simulation results based on his master thesis.

Reference:

[1] Lei Yang, Jiannong Cao, Shaojie Tangy, Tao Li, Alvin T. S. Chan "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing" , 5th IEEE International Conference On Cloud Computing (CLOUD 2012), Jun 24-29 2012.

[2]Lei Yang, Jiannong Cao, Yin Yuan, Tao Li, Andy Han, Alvin Chan, " A framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing", SIGMETRICS Performance Evaluation Review 40(4): 23-32 2013.

[3] Fan Xin, A Task Allocation Method and Its Implementation for Services Using Wearable Devices in a Mobile Cloud Computing Environment, master thesis, University of Aizu. Sep 2013

[4] Zhitao Deng, Yoshinobu Kimezawa, Nariyoshi Chida, Peng LI, Zixue Cheng " A Just-in-Time Task Allocation Method for the Efficient Usage of Resources in Wearable-Mobile Computing Environment" , The 5th workshop of IPSJ TOHOKU branch,31 Jan 2014 Koriyama, Japan.