INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2627

Two-phase grouping-based resource management for big data processing in mobile cloud computing

JiSu Park[1], Hyongsoon Kim[2], Young-Sik Jeong[3] and Eunyoung Lee[4,*,†]

[1]*Department of Computer Science Education, Korea University, Seoul, Korea*
[2]*Network Planning Department, National Information Society Agency, Seoul, Korea*
[3]*Department of Multimedia Engineering, Dongguk University, Seoul, Korea*
[4]*Department of Computer Science, Dongduk Womens University, Seoul, Korea*

SUMMARY

Big data is generated from recent social network services, and distributed processing techniques have been studied to analyze it. In particular, because of the fast spread of mobile devices, a huge amount data is generated in a mobile environment. The distributed processing technologies such as MapReduce are applied to mobile devices, thanks to the improved computing power of mobile devices. However, mobile devices have several problems such as the movement problem and the utilization problem. Especially, the utilization problem and the movement problem of mobile devices cause system faults more frequently because of dynamic changes, and system faults prevent applications using mobile devices from being processed reliably. Therefore, to cope with these significant problems of mobile devices, we propose a grouping technique based on the utilization and movement rates. In our proposed scheme, mobile devices are separated into groups by cut-off points based on entropy values. We also propose a two-phase grouping method in order to reduce the overhead of group management. The experimental result shows that our algorithm outperforms traditional grouping techniques with maintaining stable big data processing and managing reliable resource.

Received 2 April 2013; Revised 4 June 2013; Accepted 2 July 2013

KEY WORDS: two-phase grouping; resource management; mobile cloud computing; big data

1. INTRODUCTION

Recently, because of the fast spread and improved performance of mobile devices, services provided in the mobile environment have been increased. A huge amount of data has been generated in the mobile environment by increased social network services, and the data is especially called *big data* because of the large volume of the generated data. Big data has the characteristics of large volume, variety of data kinds, and need for expeditious data processing and analysis [1].

Many studies have been conducted for processing big data more efficiently. MapReduce and cloud computing are considered as the most promising candidates for this purpose. Recent big data processing technologies have utilized high-performance computing devices, but adopting the distributed environment with the distributed computing model such as MapReduce is more efficient for processing big data than using a high-performance server. In other words, MapReduce is the computing model for data analysis and management, and cloud computing is the infrastructure to provide resources for big data processing. MapReduce handles large amount of data in a short time by processing data stored on a distributed file system in parallel. MapReduce consists of map function which generates the pairs of key and intermediate value and reduce function, which combines all the intermediate pairs of the same key value [2].

*Correspondence to: Eunyoung Lee, Department of Computer Science, Dongduk Womens University, Seoul, Korea.
†E-mail: elee@dongduk.ac.kr

The cloud computing has been developed from existing cluster computing, grid computing, and utility computing [3, 4]. Mobile cloud computing has been developed from mobile computing, mobile grid computing as well as cloud computing. Therefore, mobile cloud computing is the combination of mobile and cloud computing and offers a cloud computing environment through various mobile devices. Mobile devices refer all kinds of devices that have mobility, such as laptops, tablet PCs, and smart phones. Recently, because of rapid expansion of smart phones, tablet PCs and wireless communication environment such as 4G (LTE, WiBro, etc.), the computing speed of mobile devices rapidly increases [5]. The computing power of the latest mobile devices has become comparable to that of desktop computers. Therefore researchers have attempted to use mobile resources for processing big data in mobile cloud computing environments [6–9]. In particular, computing models such as MapReduce divide a job into small blocks (64MB, 128MB or so), so it makes such jobs processed on mobile devices [6]. However, in order to use mobile devices as reliable resources for processing big data, some issues must be solved in advance.

The first issue is the volatility of mobile devices. Mobile devices show the volatility problem due to their mobility and their autonomous joining and leaving of networks. As a result, the halt of performed operation, the delay of operation completion or system faults can occur. These defects degrade the reliability of performed operations; thus, the completion of the operation is not always guaranteed. In particular, Falaki *et al.* analyzed the utilization of mobile devices [10]. Their experimental results report that the utilization of mobile devices is usually less than 25% (15 min) of an hour. Therefore, by analyzing the utilization time of mobile devices should be monitored and analyzed for reliable mobile resource provision.

The second issue is the mobility of mobile devices. In order to perform a task on a mobile device, the task request and the task result should be transferred through network before/after processing the task. However, in mobile cloud environments, network disconnection occurs frequently because of the frequent movement across wireless networks, and the task request and the task result might not be transferred properly.

To resolve these problems, we propose a technique in which we classify mobile devices into groups according to the utilization rates and the number of moves of mobile devices in order to manage the stability of performing tasks. In our classification technique, the boundary values of groups are calculated based on the entropy of mobile devices. The boundary values of the proposed technique are more accurate and more stable than arbitrary or average-based boundary values, which were used in the previous researches. Experiments show that our algorithm outperforms traditional grouping techniques with maintaining stable big data processing and managing reliable resource. The experimental result of the proposed technique is presented later in this paper.

The remainder of this paper is organized as follows. In Section 2, we present related works on mobile cloud, big data, and grouping techniques. In Section 3, we describe the system architecture of our proposed system. In Section 4, our one-phase grouping technique using entropy is described, and then the two-phase grouping technique using the similarity between group is described. In Section 5, the result of performance experiments is presented and analyzed. In Section 6, we present our conclusion.

2. RELATED WORK

Big data is the technology to handle data with the purpose of the generation and accumulation of large amounts of data, data sharing and openness, data fusion between heterogeneous areas, data analysis using super computing power, and so on. The early researches on big data have studied processing techniques and analytical methods, but current studies have been conducted to solve problems and perform experiments using big data [11].

MapReduce technology is one of the most promising techniques for big data processing. MapReduce is a computation model that easily handles the resources of a highly distributed system by managing the communication between the machines, dividing the input data and scheduling execution [2]. Hadoop is known as the most popular open source implementation of Mapreduce. Matsunaga *et. al* used Hadoop for the parallel processing of blast [12]. Pedault mentioned various MapReduce techniques on IBM Big Data Platform [13], and his adaptive MapReduce algorithm

DOI: 10.1002/dac

decreases the execution time of the map task, and overhead of task startup up to 30%. Wang *et. al* proposed the use of Kepler through MapReduce by combining Kepler and Hadoop [14]. Cloud computing has been studied as the platform for distributed processing technologies. Currently, countless social network services are generating big data through mobile devices, and researches on providing services after analyzing social big data have been actively conducted. These research trends lead to the research on mobile cloud computing using mobile devices as resources.

Recent researches on mobile cloud computing can be divided into three types [15]. The first type is the thin client. With the thin client model, applications are not installed directly on a mobile device. Services such as Google's Documents and Salesforce.coms customer relationship management service are the successful examples of the thin client model. In the second type, a mobile device is used as a resource provider [16–18]. This type of mobile cloud computing is possible because the computing power of mobile devices enhances rapidly these days. Many researches have addressed the utilization of powerful mobile devices. In the recent researches including Hadoop, a task is divided into smaller blocks (e.g., 64MB), and each block is assigned to one mobile device. The size of block is small enough to be processed effectively by a mobile device. The third type of studies are focusing placing local cloudlets between the mobile devices and cloud resources in order to reduce the workload of a cloud system [19, 20]. The role of cloudlet is like a proxy between mobile devices and a cloud system. Jiwei *et al.* [20] proposed ENDA, a three-tier architecture that leverages user track prediction, realtime network performance and server loads to optimize offloading decisions. They design a greedy searching algorithm to predict user track using historical user traces and designs a cloud-enabled WI-Fi access point selection scheme. When services of the thin client model in cloud computing are provided, it is important to provide reliable cloud infrastructure without faults. In particular, the fault problem in mobile cloud computing must be resolved carefully because the mobile cloud computing has higher fault rate than traditional cloud computing. Fault occurs in mobile cloud according to the volatility and mobility of mobile devices. To cope with these volatility problems, we propose a reliable resource management method using entropy-based group classification in this paper.

Choi and Buyya [21] worked on solving the volatility problem due to users' free joining and leaving in a P2P grid environment. In this study, groups were defined using the availability and reliability. A resource was selected from a group of devices in use. However, this technique is intended only for a wired network environment, so the movement of resources was not considered. Moreover, the cut-off points that distinguished each group were chosen arbitrarily. For mobile grid environments, Lee *et al.* [17] proposed a group-based fault tolerance algorithm. In this paper, they grouped mobile devices into four groups by using the partial and full availability. However, cut-off points for dividing groups were calculated based on the expected value for the computation and communication. Therefore, this scheme is not able to cope with the dynamic information because arbitrary cut-off points. Choi *et al.* [22] proposed resource allocation considering the characteristics of mobile resources, such as network disconnection and low battery capacity. In that study, resources were divided into two groups by performance. However, again, the cut-off points that distinguished groups were arbitrary. Jeon [23] proposed a paging mechanism based on the mobility patterns of users. In that study, the users were classified by mobility pattern into high-mobility users and low-mobility users. Once again, the cut-off points that distinguished each group were arbitrary. The user locations were managed using different techniques based on the characteristics of each user. However, in the campus environment, there exists another type of users: those who exhibit patterned mobility.

3. SYSTEM ENVIRONMENT

Mobile cloud computing is the combination of mobile computing with cloud computing and offers a cloud computing environment through various mobile devices. However, because of problems such as heterogeneity among mobile devices, low network bandwidth, and highly intermittent connections, it is difficult to integrate mobile devices in mobile cloud environments directly into a cloud environment. Thus, a proxy is used to mediate between mobile devices and a wired cloud. A proxy connects a wired cloud and mobile devices and compensates limited resources and performance

DOI: 10.1002/dac

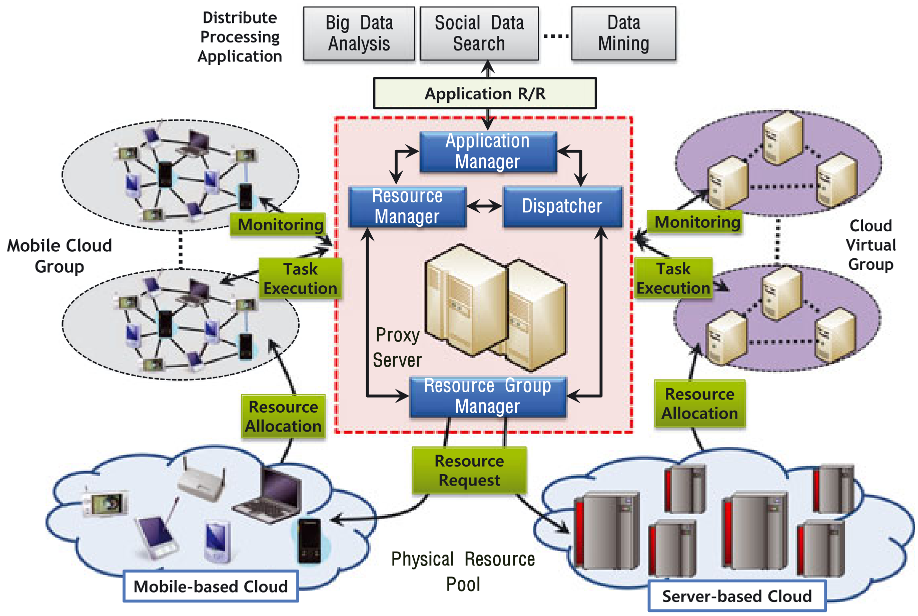


Figure 1. Mobile cloud environment.

of mobile devices; it also manages mobile devices connected to the wired cloud. Figure 1 shows the proposed proxy-based mobile cloud. For reliable resource management, the proxy server in the architecture has four components: application manager, resource manager, resource group manager, and dispatcher.

The application manager manages application programs for distributed processing such as big data analysis, social data search, and so on. It also sends the tasks to the dispatcher after separating the tasks according to the environments in which the tasks must be processed: mobile cloud environments or server-based cloud environments. The resource manager collects resource information form mobile devices through monitoring. In addition, it provides a fault tolerant scheduling for reliable resource management. The resource group manager creates groups based on utilization and movement information of mobile devices from resource manager and manages the groups. The dispatcher assigns and executes tasks to mobile cloud or server-based cloud depending on the status of the job.

4. TWO-PHASE GROUPING TECHNIQUE

4.1. Group classification using entropy

Researches on processing computation-intensive or data-intensive tasks using mobile devices has the volatility issue. The volatility causes a sudden stop of resource provision, and it leads to the halt of tasks. In particular, mobile devices have more chance of halts than servers or desktop computers.

The first reason of the volatility is the utilization rate of mobile devices. The CPUs of mobile devices are optimized for running applications rather than processing computational tasks. Therefore, the performance of CPUs of mobile devices is lower than the performance CPUs specialized for computational tasks. Mobile devices are also using multi-processing in order to enhance the response time of applications and load as many applications into memory as possible. As a result, the utilization rate of mobile devices' memory increases. The power of a mobile device's battery decreases as time goes by, and it makes the utilization rate of battery increase. We defined the utilization rate of mobile device's resource in our previous paper [24] and will use the definition for group classification. The second reason of volatility is the movement of mobile devices. In a

DOI: 10.1002/dac

mobile cloud environment, network disconnection occurs frequently because of the frequent movement across wireless networks. This interrupts the undergoing operation, and the delay or failure of completing the operation may cause a system failure.

To cope with the volatility problem of mobile devices, we propose a grouping technique for reliable management of mobile resource. The mobile device groups are created based on two factors: utilization and movement. The cut-off points of each factor are calculated, respectively, and the groups are formed according to two-dimensional cut-off points. The cut-off points for each factor are determined by calculating the entropy values. The number of groups can be changed according to the number of cut-off points of each considered factor. For example, if two cut-off points are chosen for utilization and two cut-off points are chosen for movement, nine mobile device groups will be formed for resource provision. Entropy is a measure of the uncertainty in a random variable [25]. In other words, entropy quantifies the expected value of the information. The higher is the entropy, the more uncertain has the random variable. Therefore, lower entropy means greater predictability for a random variable. The entropy of a group is measured as follows:

$$E(D_i) = -\sum_{j=1}^{k} P_i(G_j)\log_2 P_i(G_j) \tag{1}$$

where $P_i(G_j)$ is the probability of a group G_j in dataset D_i. The entropy of the whole groups is calculated from the entropy values of individual groups as follows:

$$E(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times E(D_i) \tag{2}$$

In our proposed scheme, utilization is related to the possibility of faults because the possibility of faults becomes high as utilization becomes high. We model numerical values of available utilization of resource by using the possibility of system faults. The fault rate of a typical mobile system is often modeled using the exponential distribution; the fault rate in this model shows a gradual increase over time. The fault rate of every mobile device is usually calculated using the exponential distribution model. The utilization rate for the entropy calculation consists of the number of faults of a mobile device.

To separate mobile devices into groups based on utilization, cut-off points should be determined after calculating entropy values. For a number of faults, f and a number of groups, k, there exist ${}_{f-1}C_k$ sets of cut-off points in total. We call each set of cut-offs an *instance*. After calculating the entropy of every instance of utilization, we choose the lowest entropy for utilization management as the cut-off instance. In the case of movement, the movement of a mobile device is related to the number of moves. The same process explained previously is also applied to obtain the cut-off instance for movement management.

Figure 2 shows an example of group classification according to the utilization and movement. The *y*-axis represents the utilization, and *x*-axis represents the movement. $\triangle CP_{a1}, \ldots, \triangle CP_{aL}$ are the cut-off points for utilization groups, and $\triangle CP_{m1}, \ldots, \triangle CP_{mN}$ are the cut-off points for movement groups. Therefore, $L \times N$ groups are formed using L cut-off points of utilization and N cut-off points of movement. The group U_1M_1 is the most reliable resource group with the lowest utilization and the lowest movement. On the contrary, The group U_LM_N is the least reliable resource group with the highest utilization and the highest movement. Algorithm 1 shows how to calculate the entropy values of the groups.

4.2. *Two-Phase group classification*

One-phase grouping technique calculates cut-off points to distinguish groups using entropy. The number of groups will increase as cut-off points of each factor increases.

If n cut-off points are obtained for each factor, the algorithm will generate $(n+1)^2$ groups.

For reliable resource provision, several conditions should be checked before determining which group is chosen for resource provision and scheduling the resources of the chosen group: this can

DOI: 10.1002/dac

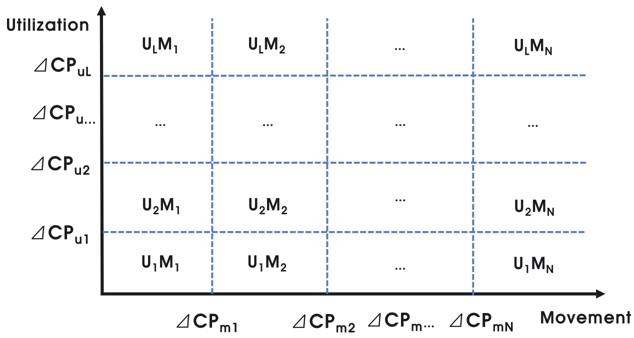


Figure 2. Example of group classification.

Algorithm 1: Entropy-based grouping algorithm

```
function create-group()
    Collect dataset from APs; // Call monitoring(MDs)
    for i = 1 to all dataset do
        Calculate entropy(D_i)
    for i = 1 to all instances do
        Sort in ascending order of entropy(D_i)
    Create groups using the lowest instance of entropy;
function entropy(dataset)
    int nnf, nf, sum; //nnf:number of not fault, nf:number of fault
    int ncp; //number of cut-off point
    for i = 0 to all data do
        double entropy, entropy-nf, entropy-f;
        for i = 0 to ncp do
            Calculate nnf, nf, sum;
            Calculate p-nnf, p-nf;    // probability of nnf and nf, p-nnf=nnf/sum
            if p-nf == 0 then
                entropy-nf = 0;
            else
                Calculate entropy of p-nnf;    // entropy-nf = (-p-nnf)*Math.Log(p-nnf, 2);
            if p-nnf == 0 then
                entropy-nnf = 0;
            else
                Calculate entropy of p-nf;    //entropy-f = (-p-nf)*Math.Log(p-nf, 2);
            Calculate entropy;    //entropy = entropy-nf + entropy-f;
            Calculate entropy-total;    // entropy-total += sum/(total-MD*entropy);
```

be the overhead of grouping. The overhead of group management usually increases as the number of groups increases. On the contrary, the differentiation between groups decreases as the number of groups decreases. The number of groups must be determined based on the characteristics of a given mobile cloud environment and target application tasks. We analyzed the performance of our grouping technique in Section 5 with various numbers of groups.

In order to compensate the overhead of group management, we propose a two-phase grouping technique. In the second phase, groups from the first phase are integrated according to the distance between groups. The group integration is performed as follows: first, the number of groups for the

DOI: 10.1002/dac

two-phase grouping is chosen. After deciding for the number of two-phase groups, the distance of each one-phase group from the origin (the optimal case) is calculated. The closer to zero is the distance of a one-phase group, the more reliable is the group.

Formula 3 shows how the distance of a one-phase group is calculated. The term *util* and the term *move* are the marginal values of each group's scope, respectively. Therefore, the number of groups for each factor is three when two cut-off points are used, and *util* or *move* has a range of 1–3, respectively. The term *ng* represents the number of groups for two-phase grouping.

$$d = \frac{\sqrt{\left(\frac{util}{ng-1}\right)^2 + \left(\frac{move}{ng-1}\right)^2}}{2} \tag{3}$$

$$(1 \leqslant util \,\&\, move \leqslant ng - 1, 0 \leqslant d \leqslant 1)$$

The distance is calculated in order to classify a group from a specific criterion point (the optimal case). In this paper, a specific criterion point was set to 1, the highest value of the distance. Therefore, the closer to zero is the distance of a one-phase group, the more reliable is the group because a short distance implies a low utilization rate and a low movement rate. In Figure 2, the shortest distance would be the distance of the group $U_1 M_1$. The range of each two-phase group is set based on the chosen number for two-phase groups. As we explained previously, the group of smaller range value usually provides more reliability.

$$S_g = \frac{\sqrt{d}}{ng - 1} \tag{4}$$

Finally, the one-phase groups are integrated into the smaller number of two-phase groups according to the calculated distance values and range values. The following is our two-phase grouping algorithm.

Algorithm 2: Entropy-based grouping algorithm

```
function two-phase-grouping()
    int ng1; //number of group in 1-phase
    int ng2; //number of group in 2-phase
    for i = 1 to ng1 do
        Calculate distance between each groups;
    for i = 1 to ng2 do
        Calculate scope of groups; // scope for 2-phase groups: S_g
        for j = 1 to ng1 do
            if S_g <= d[j] then
                Assign the group(d[j]) to group[i];
```

5. EXPERIMENTS

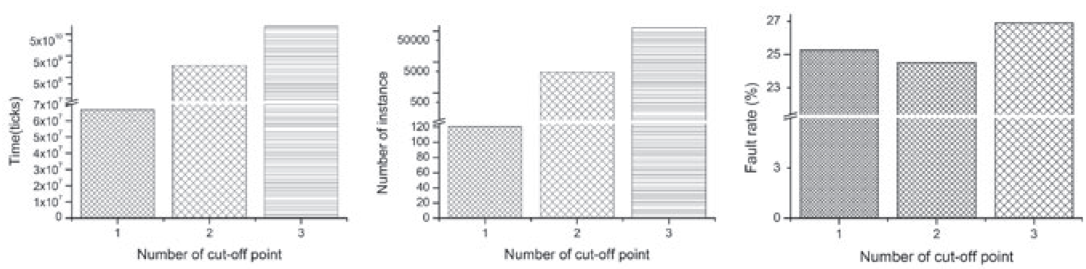
5.1. Experimental environment

We proposed the entropy-based grouping technique considering availability and mobility in our previous paper [26]. In [26], datasets are distributed with the uniform distribution on the configuration of the mobile devices and it generated nine groups using two cut-off points for each factor. In order to show the strength of our proposed algorithm, we simulated our group classification technique by using various numbers of cut-off points in various distribution environment of mobile devices.

DOI: 10.1002/dac

Table I. Configurations for simulation.

DataSet	• Fault rate : exponential distribution model • Number of mobile devices : 5000 • Number of moves : 2 30 (*exp. dis.*) • Availability : 0 1 (random) • Distribution of mobile devices (β *dis.*) - *Dataset* 1 (1.2(α) : 3 (β)) - *Dataset* 2 (1.7 :3) - *Dataset* 3 (3 :1.2) - *Dataset* 4 (3 :1.2) - *Dataset* 5 (3 :3) - *Dataset* 6 (*uniform*)
Task	• Task type : region classify for social network analysis • Number of tasks : 500,000 • Task length (time) : short (5 min)
Simulation Parameters	• *Average* fault rate of processing : 0.2 • Task processing algorithm : *FCFS*, utilization priority with group • Comparison of time according to the number of cut-off point and the number of instance - 1 cut-off point (1-cut), 2 cut-off points (2-cut), 3 cut-off points (3-cut) • Comparison of execution time in each distribution - *FCFS* : 1-cut, 2-cut - Group priority : 1-cut-p, 2-cut-p



(a) Time (b) Number of instance (c) Average fault rate of groups

Figure 3. Result of cut-off point calculation.

Table I shows the configurations for the simulations in this paper. In the experiments, we assumed that the number of moves was limited to the range 2–30 times based on the empirical research on a university campus environment. A move occurs when a mobile device changes its connection from one AP to another AP. In addition, we use the β-distribution to simulate non-uniform distribution of movements of mobile devices; various shape factors are used in order to make the distribution change.

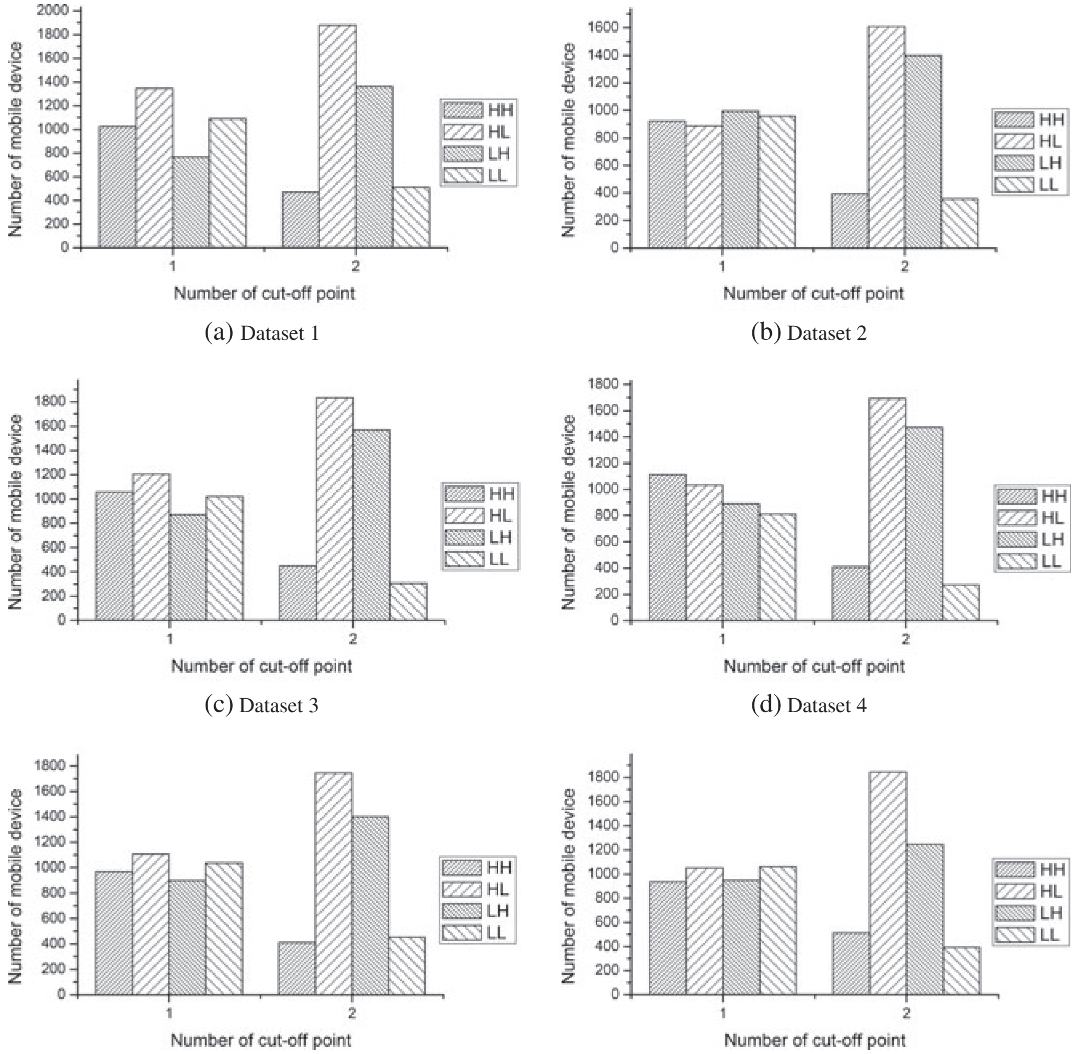
5.2. *Result of two-phase group classification*

In this experiment, we compared the time for calculating entropy by using two-phase group classification technique with one to three cut-off points for each factor. Figure 3 shows the result of data with uniform distribution (Dataset 6). The time for obtaining cut-off points is compared in Figure 3(a). The number of instances is shown in Figure 3(b).

In Figure 3(a) and (b), the y-axis represents the time, and x-axis represents the number of cut-off points. In the case of one cut-off point, 121 instances were generated, and it took 66893826 ticks for calculating entropy. On the other hand, in the case of two cut-off points and three cut-off points, 4656 and 133042 instances were created; it took 3454057561 and 225873259212 ticks, respectively. It shows that the number of instances and the calculation time increase exponentially as the number of cut-off points increases linearly.

Figure 3(c) shows the average of fault rate of each group. Figure 3(c) shows the fault rate of two cut-off points (24.59%) is lower than that of one cut-off point (25.23%) or that of three cut-off points

DOI: 10.1002/dac



(a) Dataset 1 (b) Dataset 2

(c) Dataset 3 (d) Dataset 4

(e) Dataset 5 (f) Dataset 6

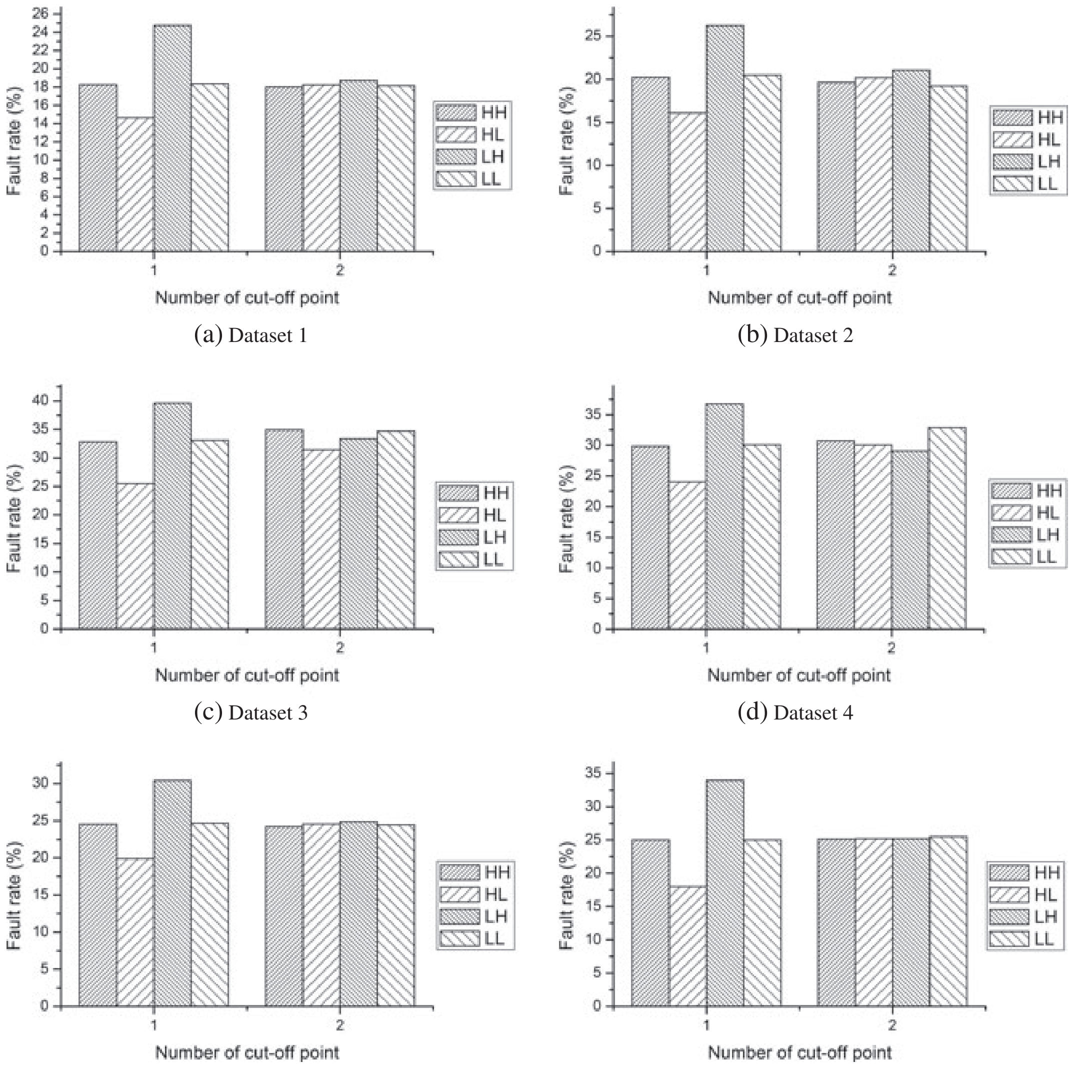
Figure 4. Number of mobile devices in each dataset.

(26.58%). The lower average fault rate of a group means higher reliability of group's resource. Note that the groups of two cut-off points provide more reliability than the groups of two cut-off points. That means, the smaller groups with many cut-off points do not always guarantee more reliability, and the optimal number of cut-off points depends on the distribution of underlying mobile devices. Therefore, the number of cut-off points should be chosen considering the characteristics of underlying mobile devices, and groups should be classified using the appropriate cut-off points. In the remaining of our experiment, we used one cut-off point and two cut-off points for utilization and movement, respectively, for comparison.

Figures 4 and 5 show the result of generating four groups using one-phase grouping technique with one cut-off point (*call it 1-cut*) and two-phase grouping technique with two cut-off points (*call it 2-cut*). Figure 4 shows the number of mobile devices of each generated group, and Figure 5 shows the fault rates of each generated group. In each graph, the first bar stands for the group with the lowest reliability ($U_L M_N$) because the utilization is high and the number of movement also is big. The last bar (the fourth bar of each graph) stands for the group of the best reliability ($U_1 M_1$).

Figure 4 shows the number of mobile devices using one-phase grouping with one cut-off points and two-phase grouping with two cut-off points for different underlying mobile device distribution.

DOI: 10.1002/dac



(a) Dataset 1 (b) Dataset 2

(c) Dataset 3 (d) Dataset 4

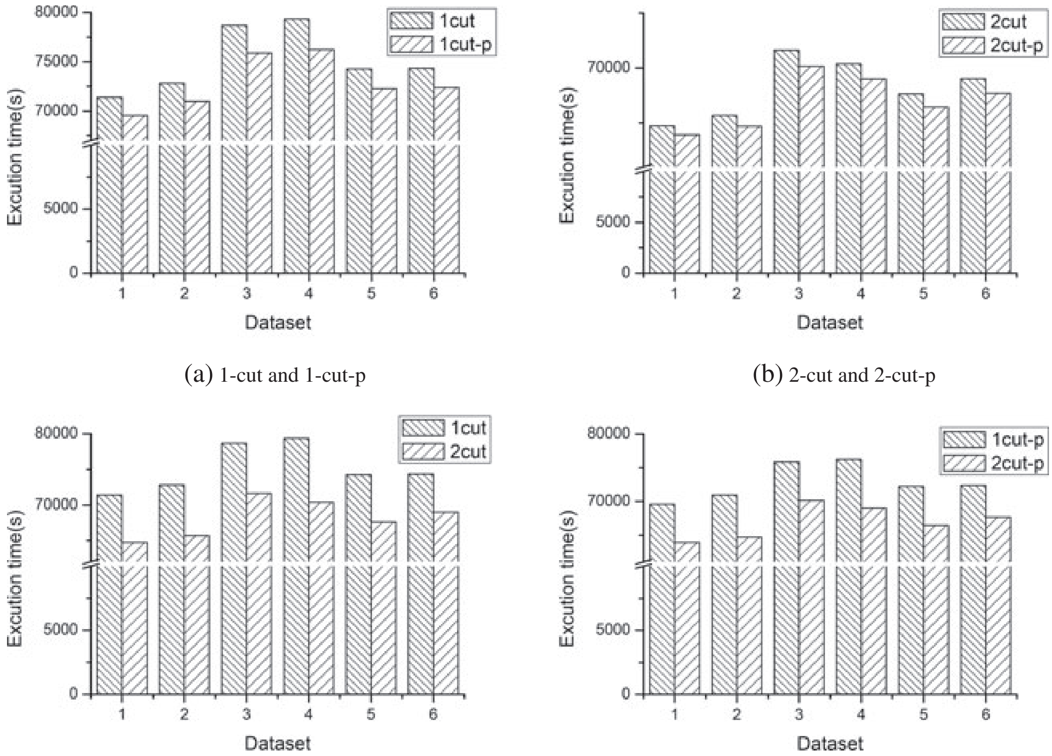
(e) Dataset 5 (f) Dataset 5

Figure 5. Fault rates in each dataset.

The graphs show that in the case of 1-cut, mobile devices of each group have been distributed to a similar size, but in the case of 2-cut, the second group and the third of group have more mobile devices than the others. Figure 5 shows the average fault rates of generated groups. In the case of 1-cut, the difference between the group of the highest fault rate and the group of the lowest faut rate goes up to 16% as in the Dataset 6 of Figure 5(f). On the other hand, in the case of 2-cut, the maximum difference between groups is 3% as in the Dataset 3 Figure 5(c). Therefore, groups that are generated by the two-phase grouping technique with 2-cut are more appropriate than groups by one-phase grouping technique with one cut-off point.

5.3. Simulated performance of two-phase grouping

Figure 6 shows the execution time of four groups generated by 1-cut or 2-cut with different dataset distributions. For selecting a group, we have used FCFS algorithm and utilization priority algorithms. In the graph, the terms *1-cut* and *2-cut* stand for the result of FCFS algorithm, the terms *1-cut-p* and *2-cut-p* stand for the result of the utilization priority algorithm. In Figure 6(a) and 6(b), we compared the performance of FCFS algorithm and the utilization priority algorithm with the



(a) 1-cut and 1-cut-p

(b) 2-cut and 2-cut-p

(c) 1-cut and 2-cut

(d) 1-cut-p and 2-cut-p

Figure 6. Comparison of number of cut-point and two algorithms.

generated group. The experimental results did not exhibit the big difference of the two algorithms. However, using the utilization priority algorithm reduced the execution time because the algorithm selects a more reliable (in other words, less utilized) group by assigning priority. Figure 6(c) and 6(d) compares the performance of our proposed techniques (1-cut and 2-cut). Experimental result demonstrates that the 2-cut groups shows better performance compared to 1-cut groups. Much shorter execution time of 2-cut groups, we analyze, is originated from the fact that 2-cut groups provide higher reliability (in other words, lower fault rate) than 1-cut groups.

6. CONCLUSIONS

In this paper, we proposed and experimented the two-phase group classification in a variety of distributions of mobile device. The existing studies that provide arbitrary cut-off points are not appropriate for mobile cloud environments, which have the volatility issue. The proposed method provides a two-phase grouping by integrating groups from entropy-based grouping with reflecting the similarity between the groups.

The experimental result shows that the algorithm generates two-phase groups successfully even though the distribution of mobile devices changes. In the first experiment, we measured the time for the entropy calculated using various number of cut-off points per factor. We also compared the average fault rates of various numbers of cut-off points per factor. The result shows that the optimal number of cut-off points depends on the distribution of underlying mobile devices. Therefore, the number of cut-off points should be chosen considering the characteristics of underlying mobile devices. In the second experiment, we measured the execution time of classified groups with one-phase grouping and two-phase grouping. We also compared two grouping techniques with different group selection algorithm. The result shows that the two-phase grouping technique outperforms the one-phase grouping technique regardless of the group selection algorithm.

DOI: 10.1002/dac

In future research, we will consider a variety of factors from the configuration of the mobile devices beyond utilization or movement and will study a way to determine the most appropriate group. Dynamic grouping in which a group of a mobile device can be changed according to the dynamic resource information of mobile cloud environments will be another good topic for our future research. In addition, we will propose a method for the grouping of mobile social network [27].

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (20120003823).

REFERENCES

1. Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AH. Big data: the next frontier for innovation, competition. *Technical Report*, and productivity. Technical Report, McKinsey Global Institute, 2011.
2. Dean J, Ghemawat S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM* 2008; **51**(1):107–113.
3. Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop, 2008. GCE'08*, IEEE, 2008; 1–10.
4. Rimal BP, Choi E. A service-oriented taxonomical spectrum, cloudy challenges and opportunities of cloud computing. *International Journal of Communication Systems* 2012; **25**(6):796–819.
5. Chiu KL, Chen YS, Hwang RH. Seamless session mobility scheme in heterogeneous wireless networks. *International Journal of Communication Systems* 2011; **24**(6):789–809.
6. Marinelli EE. Hyrax: cloud computing on mobile devices using MapReduce, *Technical Report*, DTIC Document, 2009.
7. Huerta-Canepa G. Lee D. A virtual cloud computing provider for mobile devices. *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, ACM, San Francisco, 2010; 1–6.
8. Liu Y, Chen Z, Xia F, Lv X, Bu F. An integrated scheme based on service classification in pervasive mobile services. *International Journal of Communication Systems* 2012; **25**(9):1178–1188.
9. Song EH, Kim HW, Jeong YS. Visual monitoring system of multi-hosts behavior for trustworthiness with mobile cloud. *Journal of Information Processing Systems* 2012; **8**(2):347–358.
10. Falaki H, Mahajan R, Kandula S, Lymberopoulos D, Govindan R, Estrin D. Diversity in smartphone usage. *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ACM, San Francisco, 2010; 179–194.
11. Sun-Hwa H. Science big data : grand challenges. *IT 21 Global Conference*, KIPS, Seoul, 2012.
12. Matsunaga A, Tsugawa M, Fortes J. Cloudblast: Combining MapReduce and virtualization on distributed resources for bioinformatics applications. *IEEE Fourth International Conference on eScience, 2008. eScience'08*, IEEE, Indianapolis, 2008; 222–229.
13. Pednault E. Big data platforms, tools, and research. *Technical Report*, IBM Research, 2011.
14. Wang J, Crawl D, Altintas I. Kepler+ Hadoop: a general architecture facilitating data-intensive applications in scientific workflow systems. *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, ACM, Oregon, 2009; 1–12.
15. Fernando N, Loke S, Rahayu W. Mobile Cloud Computing: A Survey. Future Generation Computer Systems, 2012.
16. Black M, Edgar W. Exploring mobile devices as grid resources: using an x86 virtual machine to run boinc on an iphone. *2009 10th IEEE/ACM International Conference on Grid Computing*, IEEE, Banff, 2009; 9–16.
17. Lee J, Choi S, Suh T, Yu H. Mobility-aware balanced scheduling algorithm in mobile grid based on mobile agent. *The Knowledge Engineering Review (accepted for publication)* 2011.
18. Palazzi CE, Bujari A. Social-aware delay tolerant networking for mobile-to-mobile file sharing. *International Journal of Communication Systems* 2012; **25**(10):1281–1299.
19. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for VM-based cloudlets in mobile computing. *Pervasive Computing, IEEE* 2009; **8**(4):14–23.
20. Jiwei Li XL, Bu K, Xiao B. ENDA: embracing network inconsistency for dynamic application offloading in mobile cloud computing. *Workshop of the Second Mobile Cloud Computing in SIGCOMM 2013*, Hong Kong, 2013.
21. Choi S, Buyya R. Group-based adaptive result certification mechanism in desktop grids. *Future Generation Computer Systems* 2010; **26**(5):776–786.
22. Choi SK, Cho IS, Chung KS, Song B, Yu HC. Group-based resource selection algorithm supporting fault-tolerance in mobile grid. *Third International Conference on Semantics, Knowledge and Grid*, IEEE, Xian, 2007; 426–429.
23. Jeon WS, Jeong DG. Design of a paging scheme based on user mobility classes for advanced cellular mobile networks. *Journal of the Korean Institute of Information Scientists and Engineers* 2002; **29**(3):216–223.
24. Park J, Yu H, Chung K, Lee E. Markov chain based monitoring service for fault tolerance in mobile cloud computing. *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*, IEEE, Biopolis, 2011; 520–525.

25. Shannon CE. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 2001; **5**(1):3–55.
26. Park J, Lee E. *Entropy-Based Grouping Techniques for Resource Management in Mobile Cloud Computing*, Ubiquitous Information Technologies and Applications. Springer, 2013; 773–780.
27. Yang K, Cheng X, Hu L, Zhang J. Mobile social networks: state-of-the-art and a new vision. *International Journal of Communication Systems* 2012; **25**(10):1245–1259.

AUTHORS' BIOGRAPHIES

JiSu Park received his BS and MS degrees in Computer Science from Korea National Open University, Korea in 2003 and 2005, respectively and PhD degree in Computer Science Education from Korea University in 2013. He is currently a Research Professor in the Department of Computer Science Education at Korea University, Korea. His research interests are in mobile grid computing, mobile cloud computing, cloud computing, distributed system, and computer education.

Hyongsoon Kim received his PhD degree in computer science at Korea University in 2009. He has been working as a researcher and a policy adviser in the field of Information and Communications Technology (ICT) infrastructure in Korea since 1997. He is currently responsible for Giga Internet pilot projects, Rural BcN (Broadband Convergence Network) projects, and Korea Research Network in National Information Society Agency (NIA) of Korea. He managed the pilot projects on IPv6, BcN (Broadband Convergence Network, Next Generation Network) and also worked as the chair of Service Transition Working Group of IPv6 Forum Korea. His research area includes mobile cloud computing, developing the ICT infrastructure policy development, and next generation network.

Young-Sik Jeong is a professor in the Department of Multimedia Engineering at Dongguk University in Korea. His research interests include cloud computing, mobile computing, IoT (Internet of Things), and wireless sensor network applications. He received his BS degree in Mathematics and his MS and PhD degrees in Computer Science and Engineering from Korea University in Seoul, Korea in 1987, 1989, and 1993, respectively. Since 1993, he has been serving as an IEC/TC 100 Korean Technical Committee member, as the IEC/TC 108 Chairman of Korean Technical Committee, and as an ISO/IEC JTC1 SC25 Korean Technical Committee member. Also, He is an EiC (editor in chief) of Journal of Information Processing Systems, an associate editor of IJCS (International Journal of Communication Systems), an editor of JIT (Journal of Internet Technology), an associate editor of Journal of Human-centric Computing (HCIS), and so on. He is also is a member of the IEEE.

Eunyoung Lee received her BS and MS degrees in Computer Science from Korea University, Seoul in 1996 and 1998, respectively and PhD degree in Computer Science from Princeton University, USA in 2004. She is currently a Professor in the Department of Computer Science at Dongduk Woman's University in Korea. Her research interests are in software security, programming language, and compiler and distributed computing.

DOI: 10.1002/dac