# Markov Chain based Monitoring Service for Fault Tolerance in Mobile Cloud Computing*

JiSu Park, HeonChang Yu
Dept. of Computer Science Education
Korea University
Seoul, Korea
{bluejisu, yuhc}@korea.ac.kr

KwangSik Chung
Dept. of Computer Science
Korea National Open University
Seoul, Korea
kchung0825@knou.ac.kr

EunYoung Lee**
Dept. of Computer Science
Dongduk Women's University
Seoul, Korea
elee@dongduk.ac.kr

***Abstract*—Mobile cloud computing is a combination of mobile computing and cloud computing, and provides cloud computing environment through various mobile devices. Recently, due to rapid expansion of smart phone market and wireless communication environment, mobile devices are considered as resource for large scale distributed processing. But mobile devices have several problems, such as unstable wireless connection, limitation of power capacity, low communication bandwidth and frequent location changes. As resource providers, mobile devices can join and leave the distributed computing environment unpredictably. This interrupts the undergoing operation, and the delay or failure of completing the operation may cause a system failure. Because of low reliability and no-guarantee of completing an operation, it is difficult to use a mobile device as a resource. That means that mobile devices are volatile. Therefore, we should consider volatility, one of dynamic characteristics of mobile devices, for stable resource provision. In this paper, we propose a monitoring technique based on the Markov Chain model, which analyzes and predicts resource states. With the proposed monitoring technique and state prediction, a cloud system will get more resistant to the fault problem caused by the volatility of mobile devices. The proposed technique diminishes the volatility of a mobile device through modeling the patterns of past states and making a prediction of future state of a mobile device**

***Keywords-component; Monitoring, Monitoring Time Interval, Mobile Cloud Computing, Markov Chain, Pattern***

## I. INTRODUCTION

Mobile cloud computing offers 'pay-as-go' cloud computing environment with various mobile devices that support mobility. Mobile devices refer all kinds of devices that have mobility, such as laptops, PDAs, tablet PCs, and smart phones. Previous mobile devices were notorious for restricted battery power and low CPU performance. However, the computing power of the latest mobile devices is getting as fast as that of desktop computers. The battery capacity is also growing, and the number of users who use mobile devices is rapidly increasing. Especially, more people use mobile devices regularly in campus or in office than ever. This trend leads researchers to try to utilize mobile devices in cloud computing.

Researches on utilizing mobile devices in mobile cloud computing can be categorized into mobile devices-as-interface and mobile devices-as-resource. Most of previous researches in mobile cloud computing have been focused on utilizing mobile devices as interface. The research on utilizing mobile devices as resources in mobile cloud environment gains attentions recently, because the population of smart phone or other mobile device users grows fast. In order to use mobile devices as resource, several problems must be solved, such as unstable wireless connection, limitation of power supply, low communication bandwidth and frequent location changes. Because a join or a leave of a mobile device is unpredictable, the undergoing process can be also interrupted unpredictably. This interruption causes the delay of operation completion, and could lead a system to a fault. Therefore, operations on mobile devices are not guaranteed for completion. This reduces the reliability of mobile devices and prevents mobile devices from being used as resource. Therefore, the dynamic characteristics of mobile devices must be considered and solved, in order to guarantee the stable usage of mobile devices as resources. In order to solve the above problems, previous researches focused on fault tolerance techniques.

Resource scheduling and fault tolerance techniques calculate state information through monitoring resource information. But, if correct resource information is not provided timely, the incorrect information would cause an accuracy problem. Therefore, a monitoring scheme that can collect and analyze dynamic state information is required in order to ensure the stable participation of resources. Monitoring schemes need to be adaptive dynamically in real time in order to monitor correct state information and reflect characteristics of mobile resources.

In this paper, we propose a monitoring technique based on Markov chain, which can analyze resource states more precisely in order to solve the fault problem that occurs by the volatility of mobile devices. The proposed technique can deal with the volatility of mobile devices by modeling the patterns of operations performed in the past and predicting the type of future operation states. The predicted information is used for fault tolerance, and it improves the reliability and performance of the system.

The rest of the paper is organized as follows: Section 2 presents the related work on monitoring services. Our mobile cloud system architecture and components which is used for the suggested monitoring scheme is described in Section 3. Section 4 proposes a monitoring technique based on Markov Chain model with Viterbi algorithm. In section 5, we present a monitoring time interval rates and the accuracy of predicted values. We draw a conclusion and discuss some future work in Section 6.

* 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(2010-0005460)
** Corresponding author

978-0-7695-4338-3/11 $26.00 © 2011 IEEE
DOI 10.1109/WAINA.2011.10

## II. RELATED WORK

The monitoring services use the pull model and the push model [1][2]. In the pull model, a server send a message to a client in order to request resource information, but in the push model, resource information is sent from a client to a server according to the monitoring policy of the server. In the pull model, the monitoring overhead is relatively small, because the resource information of a client is requested whenever the resource information is needed, But this model has a long response time and is not widely used in dynamic environments because requests are sent to clients regardless of their states. In the push model, the monitoring information is statically collected, because a system administrator determines the monitoring time intervals. If the monitoring time interval is very short, the overhead of collecting monitoring information increases. The scheme, however, cannot keep correct state information in dynamic environments if the interval is very long. Therefore, we propose a monitoring scheme that can change the monitoring time intervals in dynamic environments.

Huh et al. [3] tried to determine the monitoring time interval by observing the dynamic state of resource information and this scheme is based on the push model. But the scheme utilizes only the information of CPU among various resources so it is difficult to use the scheme in the mobile cloud environment where resources change more rapidly. MDS4[4] was developed as a part of Globus project and used for monitoring and selecting grid resources. Since MDS4 is based on the pull model, it is difficult to deploy MDS4 in the dynamic mobile environment. OVIS[12] is a resource monitoring tool in the cloud computing environment. OVIS can characterize dynamically the resource and application state, and manage optimally resources based on the monitored information. OVIS uses statistical analysis to scale data collection and resource allocation. OVIS, however, is for wired environment and statistical analysis is only used to find resources of the same attribute.

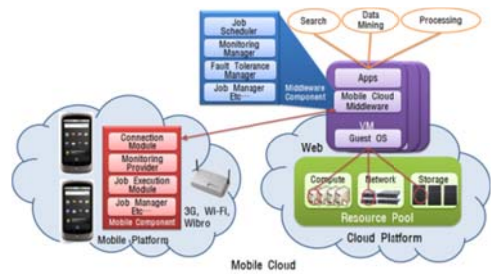
In this paper, we propose a monitoring technique based on Markov chain, which can analyze resource states more precisely in order to solve the fault problem that is caused by the volatility of mobile devices.

## III. SYSTEM MODEL

### A. Mobile Cloud System Architecture

Mobile cloud computing is a combination of mobile and cloud computing, and offers a cloud computing environment through various mobile devices. However, due to the problems such as heterogeneity among mobile devices, low network bandwidth, and highly intermittent connection, it is difficult to integrate mobile devices directly with mobile cloud environments. To mediate between mobile devices and wired grids, a proxy is used. The roles of the mobile cloud middleware are as following: supplementing insufficient performance of mobile devices, connecting mobile devices to a cloud platform, and managing mobile devices. The proposed mobile cloud architecture is shown in Figure 1. The function of each component is as follows.

The mobile cloud middleware consists of Job Scheduler, Fault Tolerance Manager, Job Manager, and Monitoring Manager. Monitoring Manager decides monitoring time intervals from the collected mobile resources information. Job Manager manages job operations in a mobile device and Job Scheduler allocates a job to mobile resources according to monitoring information. Fault Tolerance Manager predicts the fault occurrence and supply techniques such as checkpoints and replication.




cture

A mobile device consists of Connection Module, Monitoring Provider, and Job Execution Module. Connection Module manages the variety of networks to make a connection to the mobile cloud middleware. Monitoring Provider collects the mobile device's state information and sends the resources state information to Monitoring Manager. Job Execution Module runs jobs received from Job Manager and sends the processed result to Job Manager.

### B. Using Pattern of the Mobile Devices

In a mobile environment (LAN, 3G Network, etc.) network connection is mostly available, but there exist some regions where network connection is impossible such as the underground and mountainous areas. Previous researches[13][14] for mobile environments have analyzed the mobile usage in the WLAN environment, which are generally in schools or companies, and showed that there exists a usage pattern of mobile devices over time. Song and Yu[15] showed that there exists a usage pattern of mobile devices. The data was collected from the wireless network of Dartmouth University for 6 months between December 2005 and May 2006.

In this paper, we used the usage patterns of [15], whose monitoring information was acquired from the mobile devices of a university campus. The following figure shows the utilization pattern that our research team analyzed.

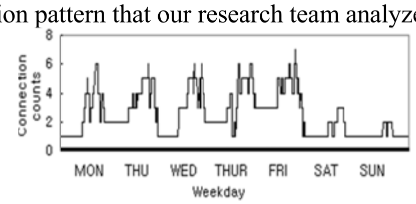



Figure 2. Connection count for a week

Figure 2 shows that the connection counts are almost the same during the weekdays, but are decreasing in the weekends.

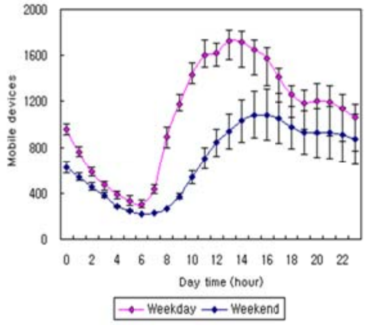



Figure 3. Average connection counts of mobile devices

Figure 3 shows the number of connected mobile devices over a week. The graph also shows the difference of connections between weekdays and weekends.

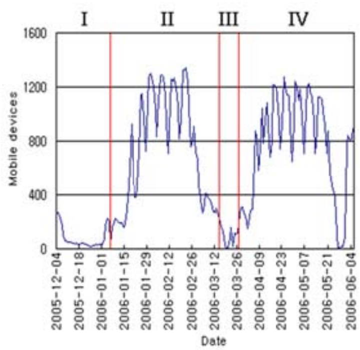



Figure 4. Connection counts of mobile devices for 4 terms

Figure 4 shows that the number of users is decreasing in the vacation period (Ⅰ, Ⅲ), and increasing in the semesters (Ⅱ, Ⅳ).

## IV. MARKOV CHAIN MONITORING MODELING

### A. Resource Information Definition

CPU power, memory, network bandwidth, and location are the examples of dynamically changing resources information of mobile devices. The information collected from mobile devices is used to calculate the utilization rate of each mobile cloud resource. The utilization of CPU and the utilization of memory are calculated by the following formula.

$$C_{util} = \frac{C_{user} + C_{sys}}{C_{total}} \times 100$$

$$M_{util} = \frac{M_{user} - M_{cache}}{M_{total}} \times 100$$

The subscript *'util'* denotes the utilization rate of a resource, *'user'* denotes the utilization rate of a user, *'sys'* denotes the utilization rate of a system, '*cache'* denotes the utilization rate of cache memory, *'total'* denotes the maximum available utilization rate.

Network Bandwidth means a remaining bandwidth calculated by subtracting the current network traffic usage from the maximum available bandwidth. It changes according to the network traffic usage over a fixed interval. Therefore, Network Bandwidth information is used to calculate the utilization rate of the network bandwidth ($N_{util}$).

$$N_{util} = \frac{(N_{total} - N_{bw})}{N_{total}} \times 100$$

$$N_{bw} = \left| \frac{1}{T} \int_{t}^{T+t} \big(C - \lambda(t)\big) dt \right|$$

$N_{bw}$ is an available bandwidth at time *t*, C is a capacity of the maximum available bandwidth, and λ is a network traffic.

Location information uses GPS information and the distance is calculated by subtracting the value of current location from the value of AP (Access Point) location. The value stands for the distance between the center of AP area and the current location.

$$L_{per} = \frac{L_{cen} - L_{cur}}{L_{lim}} \times 100$$

$$L_{cur} = \sqrt{(D_{lat})^2 + (D_{long})^2 + (D_{sp})^2}$$

$L_{per}$ is the rate of distance between the center of AP area and the current location, $L_{lim}$ is the communication coverage of distance of AP, $L_{cen}$ the center of AP of current area and $L_{cur}$ is the current location of resource.

### B. Markov Chain Modeling for Predicting Faults

Monitored state information can be classified into 3 categories by the possibility of fault occurrence. Each state is defined as follows: an operation-available and no-fault state (Stable State), an operation-available and possible-fault state (Unstable State) and an operation-unavailable state due to faults or network disconnection (Disabled State). Three state values are set based on CPU utilization rate ($C_{util}$), this classification comes CPU Utilization Guide of IBM [16].

$$S_{st}: \ 0\% \le C_{util} \le 70\%$$

$$U_{st}: 70\% < C_{util} \le 90\%$$

$$D_{st}: 90\% < C_{util}$$

Markov Chain Model (MCM) is usually defined as a matrix shown in Figure 5. In matrix *P* in Figure 5, state $S_{st}I$ means Stable State at time *I*. Therefore, the possibility of each state at time I can be written as $P_{S_{st}I}$, $P_{U_{st}I}$ and $P_{D_{st}I}$ respectively.

$$P = \begin{array}{c} \\ \\ \\ \end{array} \begin{array}{ccc|c} S_{st}J & U_{st}J & D_{st}J & \\ P_{S_{st}IS_{st}J} & P_{S_{st}IU_{st}J} & P_{S_{st}ID_{st}J} & S_{st}I \\ P_{U_{st}IS_{st}J} & P_{U_{st}IU_{st}J} & P_{U_{st}ID_{st}J} & U_{st}I \\ P_{D_{st}IS_{st}J} & P_{D_{st}IU_{st}J} & P_{D_{st}ID_{st}J} & D_{st}I \end{array}$$

Figure 5. Matrix Table *P*

$P_{S_{st}IS_{st}J}$ means the probability of transition from $S_{st}$ at time $I$ to $S_{st}$ at time *J*. And the matrix in Figure 5 must satisfy the next time condition

$$\sum_{J=1}^{m} P_{1J} = 1, \dots, \sum_{J=1}^{m} P_{mJ} = 1$$

$$P_{IJ} \geq 0, \sum_{J=1}^{\infty} P_{IJ} = 1, I = 1,2,\dots,n$$

Therefore, the state transition probability from time *J* to time *I* can be calculated as follows.

$$P_J = \sum_{n=1}^{m} (P_{nI} \times P_{nI-1})$$

$m$ is the number of states for $P_I \geq 0$ and $\sum_{n=1}^{m} P_{nI} = 1$.

The prediction of probability of the next state needs the probability of the previous state. The data of the previous state is used to calculate the probability of the next state. The information from the previous state is also used for an initial state probability. An initial state probability is calculated as follows.

$$\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$$

$\pi_i$ is an initial state probability of a state i (if $\pi_1 = P_{S_{st}}$, then $P_{S_{st}}$ is an initial state probability of $S_{st}$) and $m$ is the number of states for $1 \leq i \leq m$ and $\sum_{i=1}^{m} \pi_i = 1$.

### C. Transition Probability Estimation

If previously recorded data is insufficient, there are two methods to estimate transition probability. One is to allocate subjective values and another is to yield transition probability using statistical data. In this paper, the information of the previous state is collected from the data of the same day a week ago. We use the maximum likelihood function to yield transition probability from collected data.

$$P_{ij} = v_i, \;\; P_{i(j+1)} = 1 - v_i$$

$$L = \prod_{i=0}^{N-K} v_i^{x_i}(1 - v_i)^{y_i}, \qquad v_i \in (0,1)$$

$v_i$ is the transition probability($P_{S_{st}IS_{st}J}$) that any state at time *I* transits to any state at time *J*. $x_i, y_i$ is number that each transition occurs. For example, $x_i$ is number to transition from $S_{st}$ to $U_{st}$ and $y_i$is number to transition from $S_{st}$ to $D_{st}$. $K$ indicates the number of past states, on which the value of current state depends. Estimated value that effectively reflects *N* past patterns can be obtained by the transition probability value of the biggest *P* value.

### D. Accuracy Improvement

Failure occurrence can be predicted by the state probability of each monitoring interval. In order to improve the accuracy of prediction, the most probable value and the optimal status path is extracted and the prediction of state transition is calculated using Viterbi algorithm [17]. The probability of all the paths on a predicted state (*ps*) can be calculated by multiplying forward calculation (*FT*) and backward calculation (*BT*). Forward calculation is as follows.

$$FT_1(i) = \pi_i b_i(ps_1), \;\; 1 \leq i \leq n$$

$$FT_t(i) = \left[\max_{1\leq j\leq n} FT_{t-1}(j)p_{ji}\right] * b_i(ps_t),$$

$$2 \leq t \leq T, 1 \leq i \leq n$$

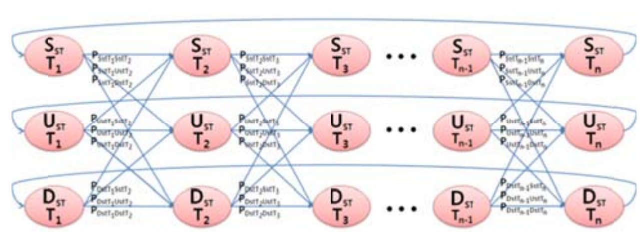
$$\tau_t(i) = \underset{1\leq j\leq n}{argmax}\left[FT_{t-1}(j)p_{ji}\right],$$

$$2 \leq t \leq T, 1 \leq i \leq n$$

$b_i$ is an arbitrary state, and τ records the optimal path at every step. *BT* is calculated by the same method as *FT*. Optimal state path is extracted as follows.

$$\hat{Q}_t = \underset{1\leq j\leq n}{argmax}\, BT_t(j)$$

$$Q_t = \tau_{t+1}\hat{Q}_{t+1}, \qquad t = T-1, T-2, \dots, 1$$



ım

$\hat{Q}$ expresses an optimal path, $Q$ is an observed state path( $q_1$ ,..., $q_t$ ). Figure 6 show a MCM using Viterbi algorithm.

### *E. Monitoring Interval*

The more jobs are processed, the more monitoring information is produced. Likewise, the fewer jobs are processed, the less monitoring information is produced. Because the monitored information changes according to resource utilization, the monitoring time interval has to be related with a job processing time. At the same time, the monitoring time interval is also related with each state because the range of resource utilization differs in states

If the monitoring time interval is short at $S_{st}$, unnecessary information is accumulated and an overhead occurs due to information collection. But, if the monitoring time interval is long, it is difficult to analyze the necessary state for resource usage or sudden faults. Therefore, if a value predicted by MCM is $S_{st}$, the monitoring time interval is calculated by job processing time and $S_{st}$'s usage limitation rate. At $S_{st}$, the monitoring time interval is called as $I_{SJTI}$ (Stable Job Processing Time Interval) and calculated by the following formulae.

$$I_{SJTI} = J_{pt} \times (1 - R_{sul}),$$

where $J_{pt}$ is *job processing time* and $R_{sul}$ is $S_{st}$'s *usage limitation rate*.

In an unstable state $U_{st}$, no one knows how resource information will change, nor which state will be the next state. The next state would be a disabled state $D_{st}$. Therefore, the monitoring time interval of $U_{st}$ must be shorter than that of $S_{st}$. At $D_{st}$, the monitoring time interval is called $I_{UJTI}$ (Unstable Job Processing Time Interval), and it is calculated by a job processing time and the usage limitation rate $U_{st}$'s and $D_{st}$'s.

$$I_{UJTI} = J_{pt} \times (1 - R_{uul}) \times R_{dul},$$

where $R_{uul}$ is $U_{st}$'s *usage limitation rate* and $R_{dul}$ is $D_{st}$'s *usage limitation rate*.

In order to calculate a monitoring time interval, other information such as battery power, which changes regardless of processed jobs, should be also considered. This interval is called $\boldsymbol{I_{STI}}$ (Static Time Interval), and it can be set by a system administrator.

## V. SIMULATION

In order for the simulation of calculating monitoring time intervals, we implemented a monitoring module using Java, JNI, SIGAR[19], RXTX[20]. The monitoring module collects the state information of CPU, Memory, Storage, Battery, Network Bandwidth and GPS. The configuration of a mobile device for collecting information is as follows; CPU: Intel Duo P8600 2.4Ghz, Memory: 4GB, Storage: 120GB, LAN: 54Mbps, GPS: BluetoothGPS. We measured the accuracy of state prediction by comparing the state information from mobile devices with the prediction using our Markov chain model and dynamic monitoring intervals.

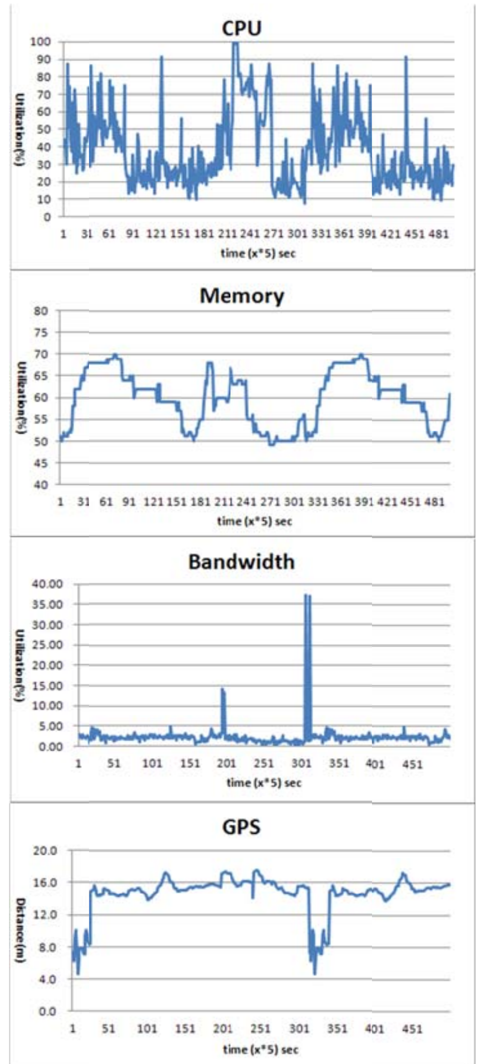



Figure 7. Monitoring information of mobile devices

Figure 7 presents the monitoring information collected from the mobile device using the monitoring module. The graphs show that the utilization rates of memory and bandwidth are quite stable, and the distance of GPS movement is relatively short. We think that it is because the resource performance of mobile devices is higher than the performance needed for job processing. In case of GPS, the distance is short because the users of mobile devices moved only within campus.

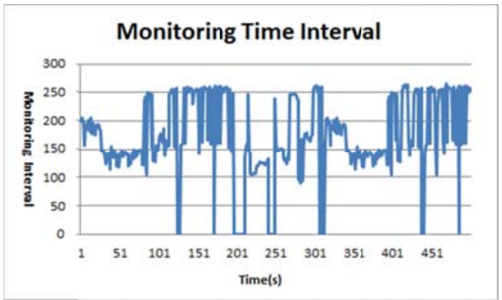


Figure 8. Monitoring time intervals

Figure 8 presents monitoring time intervals by rate. The higher the rate is, the shorter the monitoring time interval is. Job processing time is set to 300 seconds to calculate monitoring time interval.

Figure 9 shows the number of monitoring frequency according to monitoring time interval. When a static monitoring time interval is set to 60 seconds, the number of static monitoring frequency is 89 for 2,500 seconds.

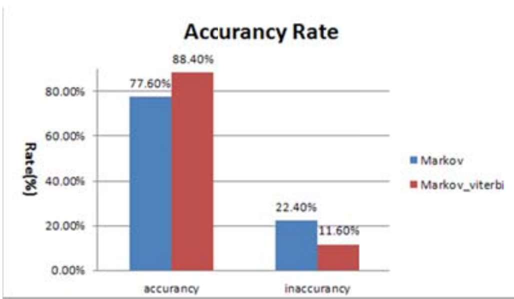
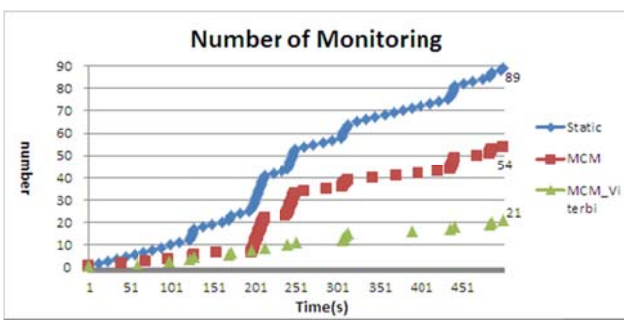


Figure 10. Accuracy rate

The graph in Figure 10 shows the accuracy of predicted states, and represents 88.4% better accuracy of predictions with comparison of basic Markov chain model.




## VI. CONCLUSION AND FUTURE WORKS

Mobile cloud computing using mobile devices as resources is considered unstable because of dynamically changing state information.

Therefore the fault tolerance must be supported for performing stable and reliable operations. Monitoring is very important since the information used to calculate the reliability for fault tolerance is provided by state information monitoring modules. To cope with the faults due to dynamic changes of mobile resource state, it is a good strategy to change the monitoring time interval dynamically.

In this paper, we proposed a technique to regulate monitoring time intervals based on our Markov chain model of mobile resource state. We applied Viterbi algorithm to basic Markov modeling, and avoided collecting unnecessary state information collection. Thus the proposed technique reduces the overhead of information collection.

For the future work, we will develop a fault tolerance algorithm using the proposed monitoring technique

## REFERENCES


[1] S Acharya, R Alonso, M Franklin, S Zdonik , "Broadcast disks: data management for asymmetric communication environments", Mobile Computing, Springer, 1996

[2] S Acharya, M Franklin, S Zdonik, "Balancing push and pull for data broadcast", ACM SIGMOD, 1997

[3] E. N. Huh, "An Optimal and Dynamic Monitoring time interval for Grid Resource Environments", ICCS'04, 2004.

[4] MDS, http://www.globus.org/toolkit/mds/ (visit 2010.5.25).

[5] GMA White Paper, http://www-didc.lbl.gov/GGFPERF/GMA-WG/ (visit 2010. 5.26).

[6] R-GMA: Relational Grid Monitoring Architecture, December 2003, http://rgma.org/ (visit 2010.5.20).

[7] R. Byrom et al.,"Fault Tolerance in the R-GMA Information and Monitoring System", EGC2005, LNCS 2470, 2005.

[8] Manvi, S.S., Birje, M.N, "Device Resource Status Monitoring System in Wireless Grids", ACEEE Intl. ACT, 2009.

[9] Manvi, S.S., Birje, M.N. "Monitoring and Status Representation of Devices in Wireless Grids", GPC 2010, LNCS 6104, 2010

[10] Sundaresanz, R., Kurcy, T., Lauriaz, M., Parthasarathyz, S., Saltz, J.: A slacker, "coherence protocol for pull-based monitoring of on-line data sources", IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003

[11] I. Foster, Y. Zhao, I. Raicu, S. Lu. "Cloud computing and Grid computing 360-degree compared", Grid Computing Environments Workshop, 2008

[12] J. Brandt, A. Gentile, J. Mayo, P. Pebay, D. Roe, D. Thompson, and M. Wong, "Resource monitoring and management with OVIS to enable HPC in Cloud computing", IEEE International Parallel & Distributed Processing Symposium, 2009

[13] B. Megdalena, & C. Paul, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network", MOBISYS, 2003

[14] H. Tristan, & K. David, A. Ilya, "The Changing Usage of a Mature Campus-wide Wireless Network", MOBICOM '04,(2004

[15] Sungjin Song, HeonChang Yu, "The Scheduling method considering the Using Pattern of the Mobile device in Mobile Grid", Journal of Korea Association of Computer Education, 2008

[16] JA Whittaker, MG Thomason, "A Markov chain model for statistical software testing", IEEE Transactions, 1994

[17] G.D. Forney Jr., "The Viterbi algorithm", IEEE Proceedings of TT, 1973.

[18] IBM, "Server Utilization Guide", http://www-903.ibm.com/kr/event/download/20080612_360 _notes/s360.pdf (visit 2010.7.5)

[19] SIGAR, http://www.hyperic.com/ (visit 2010.7.15)

[20] RXTX, http://www.qbang.org/ (visit 2010.7.20)