Adaptive and Mobility-predictive Quantization-based Communication Data Management for High Performance Distributed Computing

In Kee Kim
Sung Ho Jang
Jong Sik Lee
School of Computer Science and Information Engineering
Inha University
253, Yonghyun-Dong, Nam-Ku,
Incheon 402-751, South Korea
md10002@naver.com
ho7809@hanmail.net
jslee@inha.ac.kr

Communication data management (CDM) is an important issue in high performance distributed computing where a massive amount of data exchange frequently occurs among geographically distributed components. In this paper, we review existing CDM schemes in distributed computing systems and we propose more efficient CDM schemes. Three types of quantization-based CDM schemes are proposed: the fixed quantization-based CDM (FQ-CDM), the adaptive quantization-based CDM (AQ-CDM), and the mobility-predictive quantization-based CDM (MPQ-CDM). The FQ-CDM applies a basic theory of quantized systems to the distributed computing environment. The AQ-CDM uses a communication object clustering mechanism, which operates a pattern recognition clustering algorithm. The MPQ-CDM predicts the next states of communication objects by using past and current data and controls data communication among communication objects. The mobile object location monitoring system (MOLMS), based on High Level Architecture, is designed and developed to apply these CDM schemes to distributed computing. In this paper we conduct experiments by comparing these CDM schemes with each other on the MOLMS. The experimental results show that the AQ-CDM is the more effective scheme for communication message reduction and the MPQ-CDM is the more suitable scheme for mobile location error reduction.

Keywords: communication data management, quantized system, High Level Architecture, distributed computing, mobile object location monitoring system, high performance computing

1. Introduction

The demand for high performance distributed computing systems has increased greatly with the rapid development of network and communication technologies. Modern high performance distributed systems are not only large in scale, but also have complex system architectures. These complex and large-scale distributed systems require massive data exchange among numerous distributed components that are connected over a network. Distributed components perform computation-intensive operations when exchanging real-time data with geographical distributed data assets. Communication data management (CDM) is a critical issue in complex and large-scale distributed systems because of the limitations of real-time computation and network resources. In this paper, we pro-

SIMULATION, Vol. 83, Issue 7, July 2007 529–548

DOI: 10.1177/0037549707085438
Figures 2–4, 6, 9, 10, 13 appear in color online:
http://sim.sagepub.com

pose an effective CDM scheme based on the theory of a quantized system [1, 2] as an efficient message traffic reduction method, which is applicable for distributed computing systems. There are three types of CDM scheme: the fixed quantization-based CDM (FQ-CDM), the adaptive quantization-based CDM (AQ-CDM), and the mobility-predictive quantization-based CDM (MPQ-CDM). Each has its own unique characteristics. The FQ-CDM manages communication data among distributed components using the basic threshold mechanism of a quantized system. The other two schemes are the more improved methods. The AQ-CDM makes object clusters by considering the characteristics of the communication object, and manages communication data using the threshold mechanism of a quantized system. The threshold mechanism in the AQ-CDM is appropriate for previously formed clusters. The MPQ-CDM predicts the next mobility and location of an object using past and current data, and manages communication data with the prediction. In this paper we evaluate the system performances of the quantization-based CDM schemes with various experiments. We design and develop the mobile object location monitoring system (MOLMS), based on high level architecture (HLA) [3]. The FQ-CDM, AQ-CDM, and MPQ-CDM are all developed on the MOLMS.

This paper is organized as follows. In Section 2 we review existing CDM schemes in distributed systems. In Section 3 we introduce the three quantization-based CDM schemes and show the design and implementation of the MOLMS. The system performance of each CDM scheme is analyzed in Section 4. In Section 5 we evaluate the system performances of the three CDM schemes with experiments on the MOLMS. In Section 6 we give our conclusion.

2. Related Work

2.1 Communication Data Management

CDM is a fundamental requirement for effective system performance in a real-world distributed computing environment because of the limitations of communication and computation resources. CDM has been issued in many computation fields, such as traffic engineering. In particular, our research focuses on CDM for mobile objects (MOs) that operate with high mobility. In this section we describe the existing CDM schemes that are required for MOs. The CDM schemes for MOs can be classified into two types: region-based models [4, 5] and grid-based models [4–8]. These schemes are applied to data communication among MOs in large-scale distributed systems.

2.1.1 Region-based Model

A region-based model depends on a regional overlapping [4, 5] between sending and receiving components. A sender finds a receiver using a matching process [4, 5], which regionally overlaps between a sender and a receiver. The matching process makes a routing space [4] for message communication. After matching, a sender can communicate with a receiver. For the matching process, a location identification process between a sender and a receiver is needed. Thus, each sender and receiver continuously interchanges location messages with each other for the location identification. In a region-based model, when senders or receivers frequently change their locations, location message communication through the matching process can frequently be performed. Here, network traffic is increased and overall computation performance is decreased. In this model, the worst case of time complexity for the matching process becomes $O(n^2)$ [5].

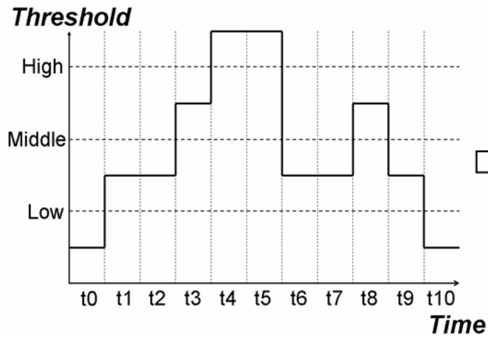
2.1.2 Grid-based Model

A grid-based model is classified into two types: the fixed grid-based model [4] and the dynamic grid-based model [5–8]. The fixed grid-based model is used to reduce communication and computation overheads from the matching process of a region-based model. The fixed grid-based model divides a routing space into grid cells with the same size [5].

Each grid cell maintains each multicast group, and a multicast group has regionally overlaid communication components. The fixed grid-based model avoids the matching process of a region-based model and constitutes matching processes between senders and receivers in each cell. Deciding on the size of a grid cell is an important issue in the fixed grid-based model [4, 5].

When the size of a grid cell is larger, a larger number of multicast groups is assigned to each cell. In this case, the communication performance of the fixed grid-based model is similar to the general multicast method and the reduction of communication traffic is decreased. However, when the size of a grid cell is smaller, a smaller number of multicast groups is allocated to each grid cell. In this case, frequent state updates of each multicast group are generated. Obviously, the fixed grid-based model minimizes the execution of matching processes between senders and receivers, and reduces communication and computation overheads. However, the fixed grid-based model has several disadvantages, as follows [5, 7]:

- when the routing space is larger, a larger number of multicast groups is needed;
- senders and receivers in the fixed grid-based model have less regional closeness than those of a region-based model because of the lack of a matching process;
- if the sender and receiver are located at the boundary of a grid cell, then superfluous meaningless messages are transmitted.



Threshold	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
High	0	0	0	0	1	1	0	0	0	0	0
Middle	0	0	0	1	1	1	0	0	1	0	0
Low	0	1	1	1	1	1	1	1	1	1	0

Figure 1. Difference between quantized results in the quantized system

To solve these problems of the fixed grid-based model, a dynamic grid-based model is proposed. In the dynamic grid-based model, multicast groups are allocated to some grid cells, while all grid cells have multicast groups in the fixed grid-based model. The dynamic grid-based model allocates a multicast group to particular grid cells that have communication components such as a sender and a receiver. The allocation of a multicast group is decided with the intersection part of sending and receiving regions. The dynamic grid-based model has two advantages [4–8]:

- reduction of needless data transmission of a sender;
- reduction of the number of multicast groups.

2.2 Overview of Quantized System

The quantization-based CDM scheme is based on the theory of a quantized system, which is described in Lee and Zeigler [1] and Zeigler et al. [2]. The quantized system [1, 2] uses a discrete value that is divided by a threshold (i.e. quantum size). If the value of the real world exists between thresholds (i.e. high, middle, and low), a threshold is considered as a quantized value. Therefore, the quantized system is a system that transforms a continuous value/state into a discrete value/state. There is a component called a quantizer [1, 2] that divides the trajectory of a continuous real value/state into that of a discrete value/state. The quantized system uses a threshold and can change an output trajectory through the adjustment of a threshold. In the case of a large threshold, the variation of an output trajectory after quantization is reduced. However, in the case of a small threshold, the output trajectory of the quantized system is similar to the variation of an input trajectory. Therefore, if the size of a threshold is close to zero, both trajectories of input and output will be the same.

Figure 1 shows the difference between quantized results in the quantized system [1, 2], according to the variation of the size of a threshold. In Figure 1, we assume that a communication object has two states, (0, 1); 0 indicates the data non-transmission state of an object and 1 is the data transmission state. The quantized system in Figure 1 has three threshold sizes: high, middle, and low. This system uses one type of threshold. This system transmits data when the state of an object changes to the other state over the threshold. If the system uses the high threshold, then the data transmission states of an object are t4 and t5 of the time series. When this system sets the size of the threshold to middle, then the data transmission states of an object are t3, t4, t5, and t8 of the time series. When the size of the threshold is small, the data transmission states of an object are t1–t9 of the time series. Therefore, in the quantized system, the decision about the size of the threshold is an important issue. In order to decide on the size of the threshold, the quantization-related features of a specified system should be noted.

In this paper, the quantized system is used for CDM schemes of complex and large-scale distributed systems. The CDM schemes are applied to the MOLMS. The MOLMS decides the movement of a MO and filters transmission data through the CDM schemes. More details are given in Section 3.

3. Quantization-based Communication Data Management

CDM plays a critical role in a large-scale and complex distributed system by improving system performance through reducing network resources for data communication. In this paper we propose quantization-based CDM schemes that reduce communication messages among distributed components. We show the three quantization-based CDM schemes: FQ-CDM, AQ-CDM, and MPQ-CDM. The FQ-CDM is a simple and uniform model, which simply applies the theory of a quantized system to CDM. The FQ-CDM is also a basic model for developing the other two models. The AQ-CDM and MPQ-CDM are the more improved models. The AQ-CDM uses several thresholds that show the movement characteristics of a MO. The MPQ-CDM controls communication messages by predicting the next state of a MO using an existing predictive model with statistics. In addition, we design

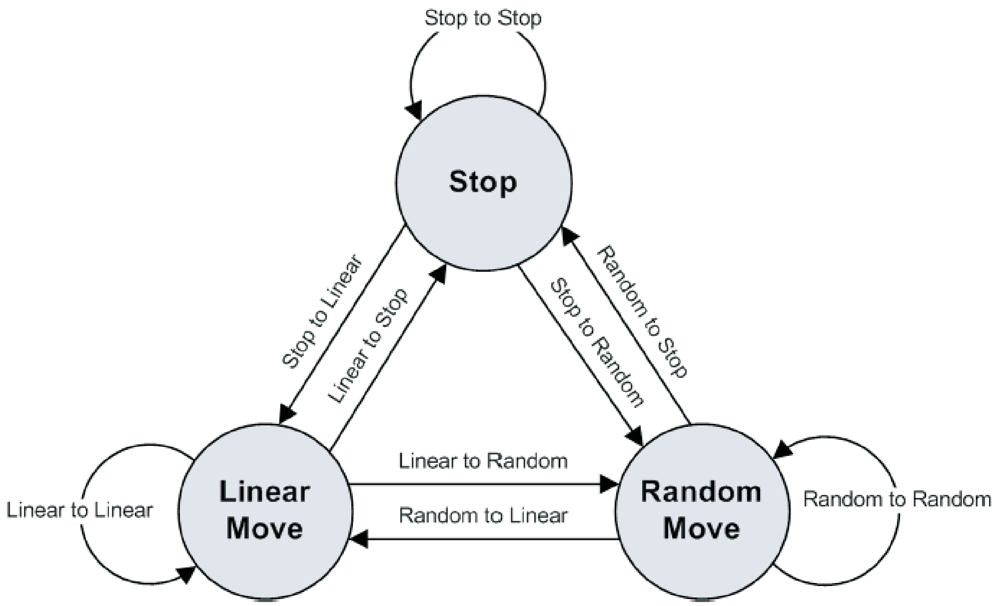


Figure 2. The mobility states of a MO

and develop the MOLMS to apply the quantization-based CDM schemes to a distributed computing application. Details of the design and implementation of the MOLMS are described in Section 3.1. Detailed descriptions of the FQ-CDM, AQ-CDM, and MPQ-CDM are given in Sections 3.2, 3.3, and 3.4, respectively.

3.1 Application: Mobile Object Location Monitoring System

Here, we describe the MOLMS. First, we model a MO to design the MOLMS. A MO has three states: linear movement, random movement, and stop [9–11]. Figure 2 shows the three states of a MO. The linear movement state is when a MO is constantly moving towards a destination. A moving car on a road is a good example. The random movement state is when a MO moves randomly in a limited area. An employee in an office is a good example. The stop state is when a MO stops moving. For example, the stop state is when a car is parked for a certain amount of time. There are more states than these three states mentioned above. However, we assume that most states can be expressed with these three states. A MO is able to change its movement state. A transition among the movement states generates the various mobility patterns of a MO. The MOLMS conducts quantization processes using a threshold based on the movement distance of a MO. The moving direction and velocity in the linear movement state are constant. In the stop state, movement of a MO does not occur. The MOLMS finds it easy to apply a threshold to a MO in the stop and linear movement states. For a MO with random movement, the velocity of a MO is between $V_{\min}$ and $V_{\max}$ and the direction of the MO is between 0 and 2π. Thus, the MOLMS finds it hard to apply a threshold to a MO in the random movement state.

The MOLMS includes four main components: agent, cluster manager, central manager, and location server.

- Agent: an agent locates a MO, which moves every second, and then transmits a message, which holds the location information of a MO, to the cluster manager.
- Cluster manager: a cluster manager integrates incoming messages and generates location information for entire regions. It then transmits the generated and integrated location information to a central manager.
- Central manager: a central manager stores all location information transmitted from numerous cluster managers to a location server and specifies the moving paths of all MOs.
- Location server: a location server stores the location information of all MOs.

In this paper, we have used the High Level Architecture (HLA) version 1.3 Specification [3] to develop the MOLMS in a distributed computing environment. The HLA is the middleware proposed by the United States Department of Defense [12] for the effective implementation of a distributed simulation environment [3]. The HLA has a simulation feature for effective message communication among distributed components. This message communication feature of the HLA is appropriate to monitor MOs

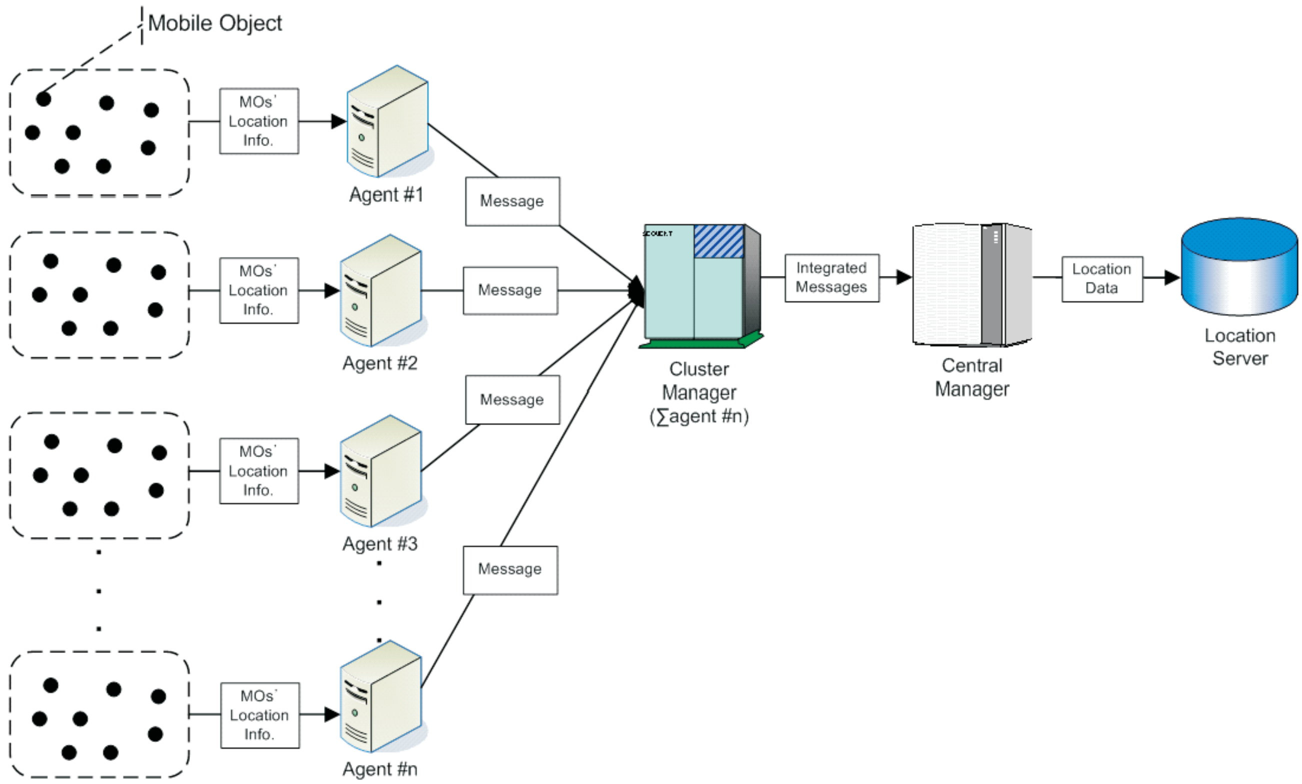


Figure 3. Architecture of the MOLMS

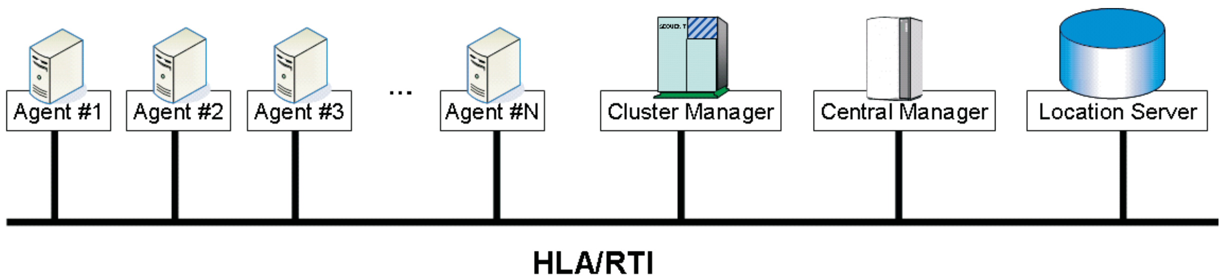


Figure 4. Implementation of the MOLMS based on the HLA middleware

that have high mobility. Figure 4 represents the implementation that shows connections among distributed components of the MOLMS based on the HLA middleware.

3.2 Fixed Quantization-based CDM Scheme

The FQ-CDM scheme is a simple and uniform CDM scheme that uses a basic quantized system. To monitor the location changes of a MO precisely, an agent must report the location of a MO to a cluster manager. However, if there are many MOs, message communication cannot be performed accordingly. Thus, for traffic reduction, the FQ-CDM has to recognize the movement of a MO using the threshold of a quantized system. Here, a threshold means to a previously assigned distance. If a MO moves more than a threshold distance, the MOLMS recognizes the movement as an effective movement. If a MO moves less than a threshold, the MOLMS does not recognize the movement. Here, if a MO moves within an area of a threshold, the MOLMS cannot pinpoint the exact location of a MO, and then a location error is generated. Deciding on a threshold is a critical issue in using the FQ-CDM. If the threshold is larger, then the number of messages is

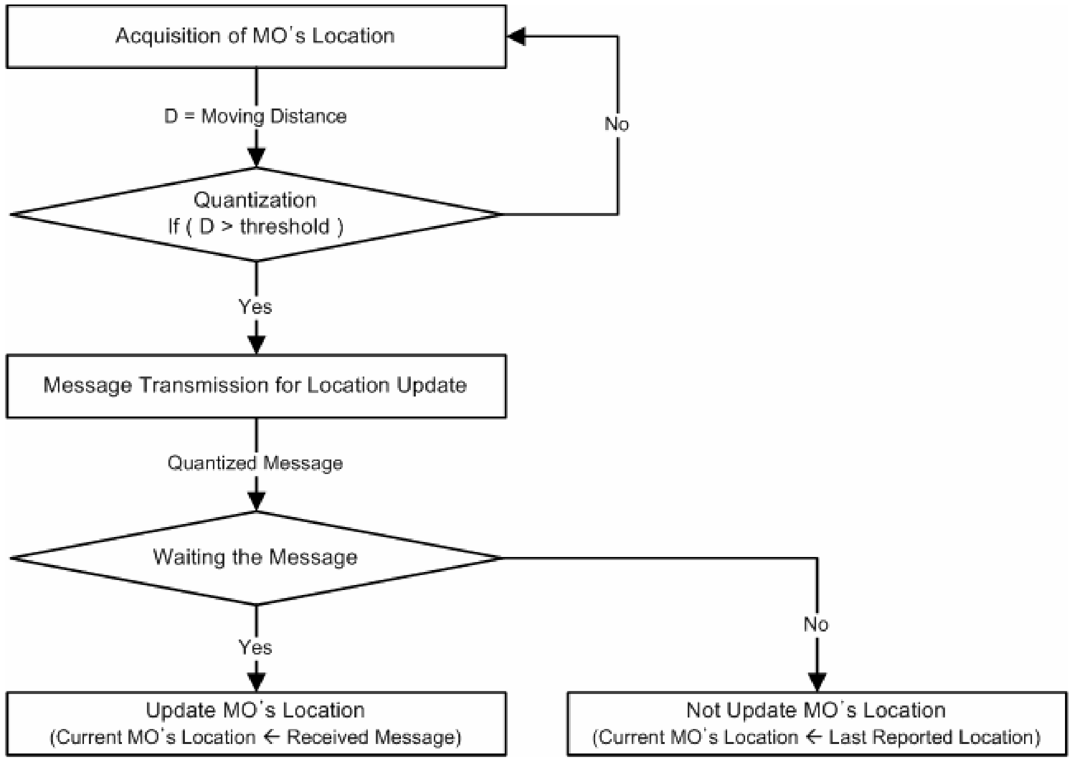


Figure 5. Process of the FQ-CDM

reduced more. Effective message traffic reduction eventually increases location errors. However, if we set a threshold to be smaller, location errors are reduced more, but effective message traffic reduction is not performed. The process of the FQ-CDM is shown in Figure 5.

The process of FQ-CDM is composed of the following five steps: (1) acquisition of a MO's location; (2) quantization; (3) message transmission; (4) waiting for a message; (5) location update.

(1) Acquisition of a MO's location. In this step, an agent acquires a MO's current location. This step calculates the moving distance from the last reported location to the current location using $SQRT\left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right]$ where x_2 and y_2 are the current coordinates of a MO, and x_1 and y_1 are the last reported coordinates of a MO.

(2) Quantization. An agent compares the size of the threshold with the moving distance of a MO. If a MO moves a longer distance than the size of the threshold, then the next step (message transmission) is executed. However, if a MO moves a shorter distance than the size of the threshold, the next step is not executed.

(3) Message transmission. If a MO's moving distance is longer than the size of the threshold, an agent transmits the MO's current location to a cluster manager.

(4) Message reception. In this step, the cluster manager receives a message from an agent.

(5) Location update. If the cluster manager receives a message, then the MO's location is updated. However, if the cluster manager does not receive a message, the MO's current location is the last reported location. Thus, the location of the MO is not updated.

Figure 6 represents the MOLMS with the FQ-CDM. An agent applies the size of the threshold to all MOs that move within a specified region. The agent monitors and quantizes the movement distances of all MOs. In this case, when a MO moves within the threshold (i.e. distance), the agent transmits a location message to the cluster manager. The cluster manager integrates these incoming location messages from all agents. The cluster manager transmits the integrated location information of all MOs to a central manager. The central manager generates moving paths through a mutual operation with a location server and stores location data.

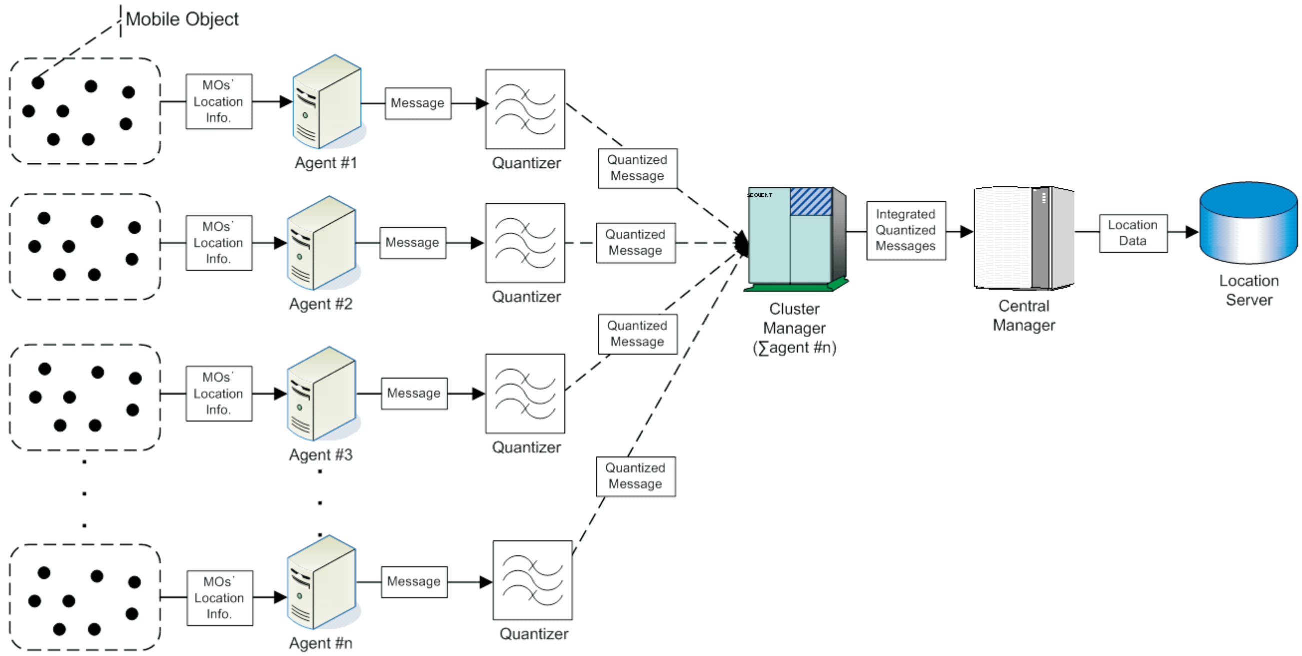


Figure 6. Architecture of the MOLMS with the FQ-CDM

3.3 Adaptive Quantization-based CDM Scheme

The AQ-CDM is an improved model of the FQ-CDM. For development of the AQ-CDM, we consider the characteristics of a MO, as mentioned in Section 3.1. The FQ-CDM is a model that applies the unit size of a threshold to all MOs. The AQ-CDM applies various sizes of a threshold to a MO. The size of a threshold depends on the mobility patterns of a MO. The AQ-CDM creates MO clusters that include MOs with similar mobility patterns. Each size of threshold suits each cluster. We expect it to be more efficient to reduce traffic messages by assigning various sizes of threshold. The AQ-CDM constructs MO clusters using main two factors of a MO's movement: velocity and direction. Figure 7 shows a mobility pattern classification of MOs based on velocity and direction.

At first, the MOLMS operates a MO classification with the moving velocity of a MO and makes n types of MO groups. Then, the MOLMS conducts a moving direction-based classification and classifies each MO group into several MO subgroups. After two classifications, MO clusters with mobility patterns are created. Each MO cluster is assigned to each agent of the MOLMS. We use the sequential clustering algorithm [13] to classify the mobility patterns of MOs. This algorithm is shown in Figure 8.

The sequential clustering algorithm for classifying and clustering MOs has three steps: (1) velocity/direction measurement; (2) clustering; (3) self-learning.

(1) Velocity/direction measurement. This step measures a similarity difference between existing clusters (C) and calculates the moving velocity and direction of MOs. $d(x, C)$ denotes a similarity difference between data x (i.e. direction and velocity) of a MO and C.

(2) Clustering. In this step, the MOLMS sets a minimum distance (α) which represents a similar difference between x and C. The MOLMS also compares $d(x, C)$ with a minimum distance. If $d(x, C)$ is less than α, then x is contained in C. However, if $d(x, C)$ is more than α, then a new cluster is created.

(3) Self-learning. This self-learning is an iteration step of velocity/direction measurement and clustering steps. The mobility pattern of a MO changes. At this time, a MO should be contained in another cluster. Thus, this is a cluster adjustment step with a change in the mobility pattern of a MO.

The reason for using MO clusters with mobility patterns is to perform a suitable quantization for MOs that have various mobility patterns. Figure 9 shows the MOLMS with the AQ-CDM scheme. The cluster manager of the MOLMS with the AQ-CDM classifies all MOs into several MO clusters. The cluster manager creates and manages MO clusters. Each MO cluster is assigned to each agent by the cluster manager. Agents continue their operation of monitoring.

The processes of the AQ-CDM consist of four steps: (1) initial classification of MO mobility patterns; (2) initial clustering; (3) recognition of the MO location via quantization; (4) cluster reconstruction.

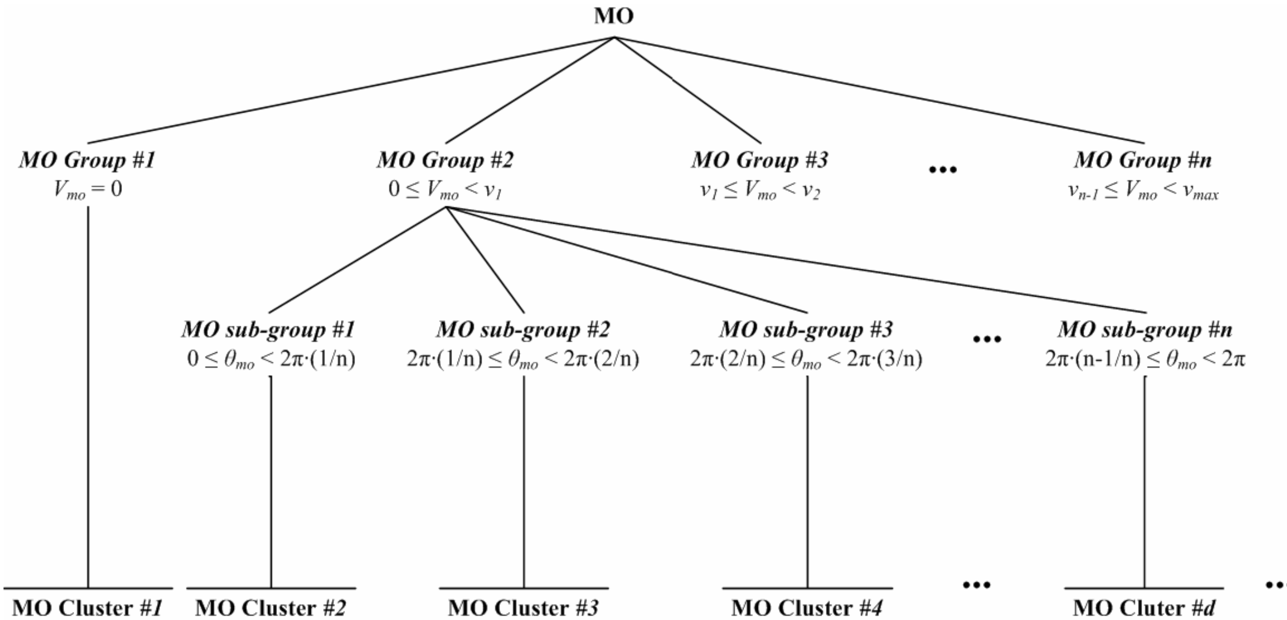


Figure 7. Mobility pattern classification with moving velocity and direction: V_{mo}, moving velocity of a MO; θ_{mo}, moving direction of a MO; v_{max}, maximum velocity of a MO

$m = 1$
$C_m = \{x_1\}$
For $i = 2$; $i <=$ sizeof(Dataset); i++)
    Find C_k: $d(x_i, Ck) = \min_{1<=j<=m} d(x_i, C_j)$
    If $d(x_i, C_k) > \alpha$ AND ($m < m_{max}$) then
        $m = m + 1$
        $C_m = \{x_i\}$
    Else
        $C_k = C_k \cup \{x_i\}$
        Update C_k
    End If
End For

Figure 8. Sequential clustering algorithm

(1) Initial classification of MO's mobility patterns. In this step, the MOLMS collects the moving characteristics of all MOs and classifies MOs into suitable clusters.

(2) Initial clustering. In this step, several clusters are created with the initially collected mobility patterns of MOs.

(3) Recognition of a MO's location via quantization. In this step, the MOLMS monitors movements of MOs and applies each threshold to each cluster in considering the characteristics of the MOs' mobility patterns. A unique threshold is applied with a unique pattern of MOs.

(4) Cluster reconstruction. Mobility patterns of MOs change as time goes by. The MOLMS reconstructs MO clusters and assigns reconstructed clusters to a cluster manager.

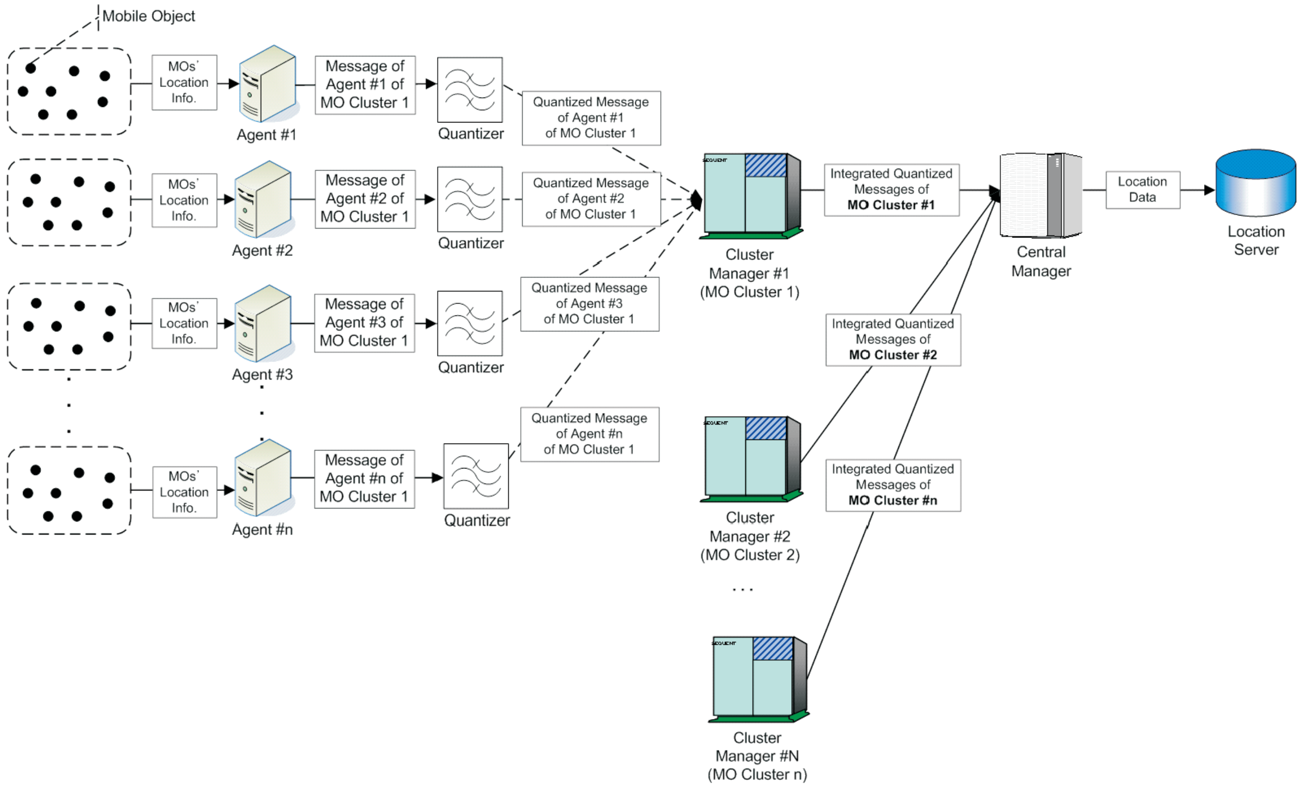


Figure 9. Architecture of the MOLMS with the AQ-CDM

Generally, the AQ-CDM can execute a more efficient message traffic reduction scheme as it considers the characteristics of mobility patterns of MOs and assigns different thresholds. However, the AQ-CDM needs additional operations of cluster construction and reconfiguration, which can increase the processing costs. Also, when a MO cluster is being constructed, there should already be a number of clusters. Thus, we need to decide on an optimal number of clusters.

3.4 Mobility-predictive Quantization-based CDM Scheme

The MPQ-CDM scheme is based on the theory of quantized system and a statistical prediction model that have complementary roles. In a concept of a general quantized system, the MOLMS has to wait for the movement of a MO and applies a threshold. After this process, the MOLMS decides whether a MO crosses a boundary or not. However, the MPQ-CDM uses a statistic prediction model to predict the mobility of a MO. The MPQ-CDM applies a threshold only if it predicts that a MO moves more than the size of the threshold.

We use the second-order exponential smoothing (SOES) prediction model [14] to predict the next mobility and location of a MO. This MPQ-CDM uses historical data. This SOES prediction model is expressed as

$$\hat{P}(d+T) = \left(2 + \frac{\alpha T}{1-\alpha}\right) S_d - \left(1 + \frac{\alpha T}{1-\alpha}\right) S_d(2),$$

$$S_d = \alpha P(d) + (1-\alpha) S_{d-1},$$

$$S_d(2) = \alpha S_d + (1-\alpha) S_{d-1}(2) \qquad (1)$$

where $\hat{P}$ is the predicted result, P is the current data, d is the current time, T is the time elapsed, α is the smoothing constant, which generally has the value of $0 < \alpha < 1$, S_d is the first-order exponential smoothing prediction model, and $S_d(2)$ is the double-smoothed statistics. Prediction accuracy is affected by the smoothing constant (α) [14]. Therefore, it is necessary to revise this prediction model by adjusting the value of the smoothing constant (α) by constantly acquiring and analyzing the location data of a MO. Both moving velocity (v) and direction (θ) must be predicted for the SOES prediction operation. Estimated velocity ($\hat{v}$) and predicted direction ($\hat{\theta}$) can be calculated using equation (1). We can obtain the next coordinates of x and y by inserting the calculated data into equation (2) [15]

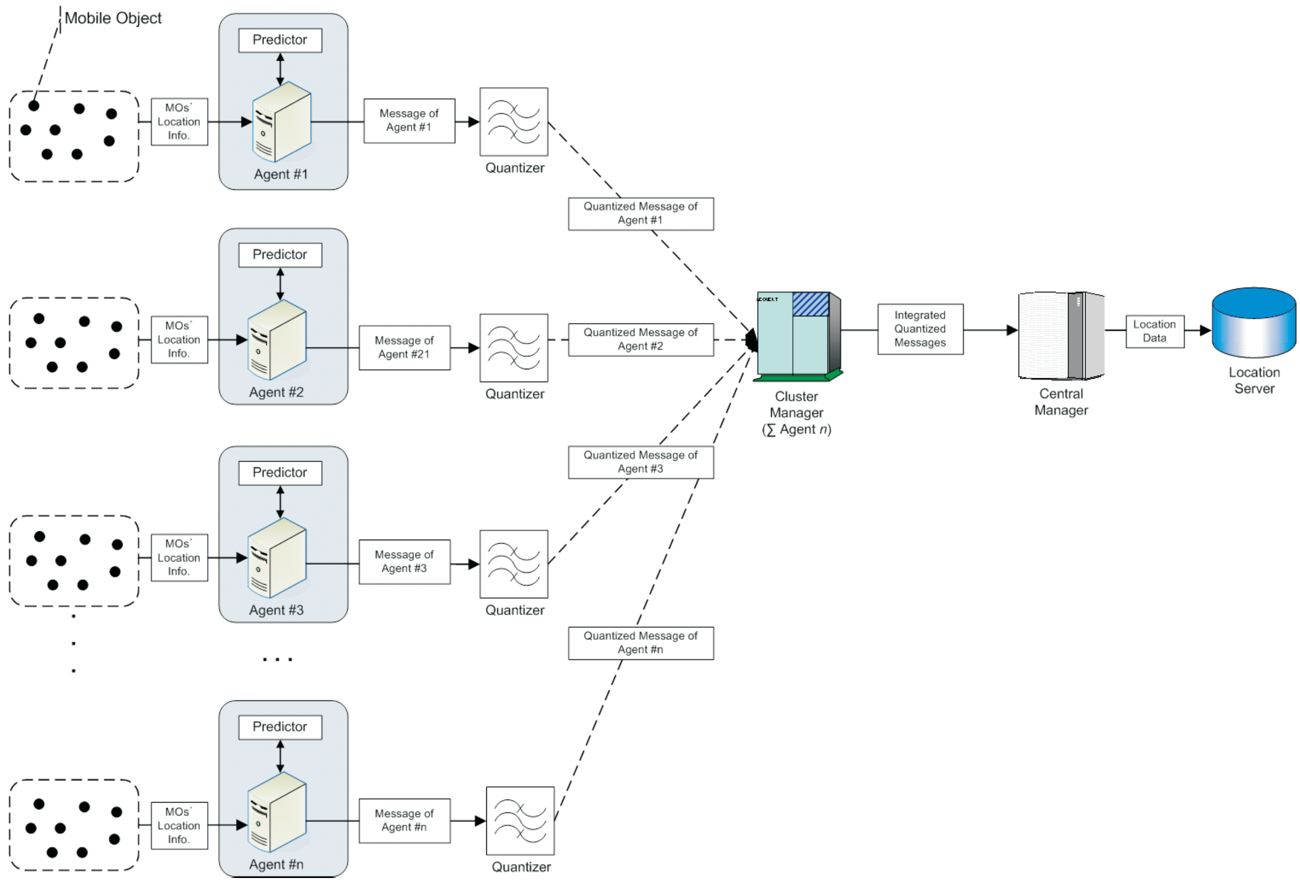


Figure 10. Architecture of the MOLMS with the MPQ-CDM

$$x_2 = x_1 + \hat{v}\cos\hat{\theta}, \quad y_2 = y_1 + \hat{v}\sin\hat{\theta}, \qquad (2)$$

where (x_1, y_1) represents the current coordinates of a MO and (x_2, y_2) are the next coordinates.

The MPQ-CDM is created to reduce message traffic in a distributed environment. The FQ-CDM generates a location error because of quantization. A quantized system decides whether to transfer location data or not by monitoring the boundary (i.e. threshold) crossing of a MO. When a MO moves within a threshold limit, the FQ-CDM assumes that the MO stops. The FQ-CDM results in a location difference between an actual location and an estimated location in the MOLMS. However, the MPQ-CDM uses the statistic prediction model to predict the next mobility and location of a MO. The MPQ-CDM estimates a more accurate location for a MO than the FQ-CDM, with a smaller location error.

The prediction process of the MPQ-CDM is composed of four steps: (1) acquisition of initial moving data of MOs; (2) prediction; (3) revision of prediction; (4) applying revised prediction.

(1) Acquisition of initial moving data of MOs. The MPQ-CDM predicts the next state mobility and location of a MO using both past and current data. Thus, the step of acquiring initial location data is required.

(2) Prediction. In this step, the MPQ-CDM predicts the next state mobility and location of a MO using the collected location data of a MO.

(3) Revision of prediction. Most mobility patterns of MOs change as time goes by. Therefore, the prediction also must be revised. This revision can make more accurate prediction possible.

(4) Applying revised prediction. In this step, the prediction, which is revised in the third step, is actually applied to the MOLMS with the MPQ-CDM.

The first step (i.e. acquisition of initial moving data of MOs) is only executed once for all MOs; however, the remaining steps are repeatedly executed.

Figure 10 shows the MPQ-CDM scheme on the MOLMS. In the MOLMS, each agent has a predictor and

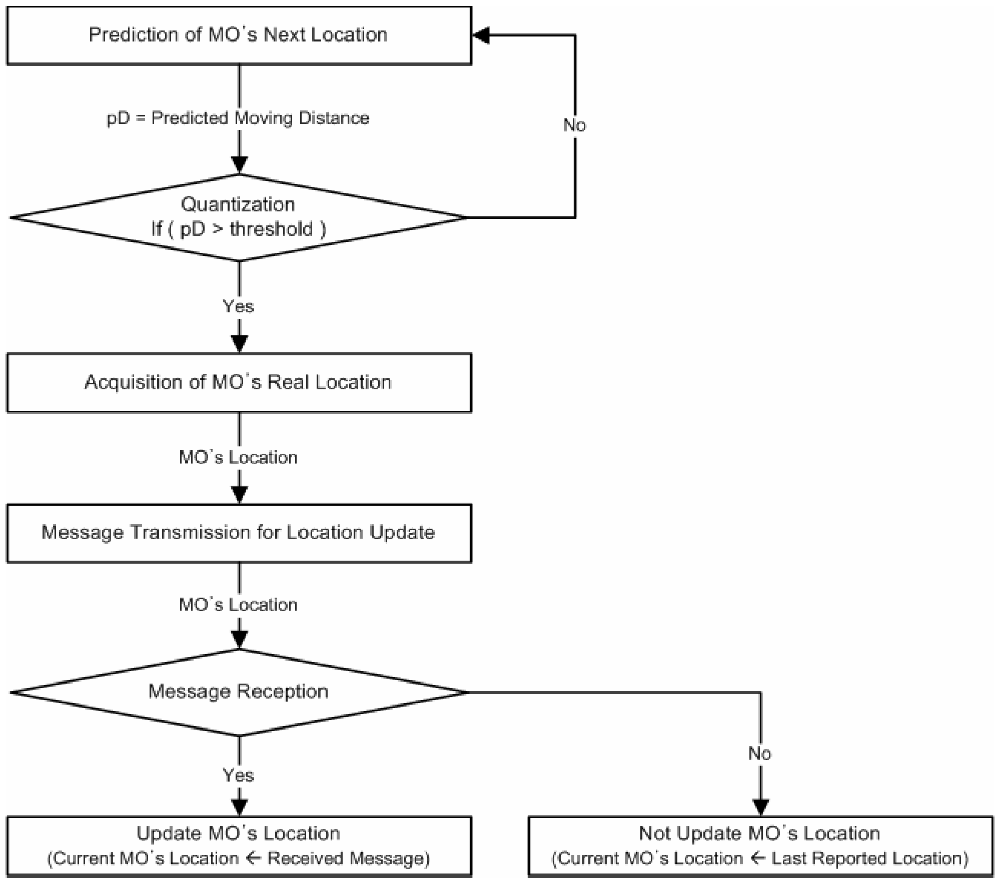


Figure 11. Process of the MPQ-CDM

a quantizer. The agent monitors a MO using its predictor and quantizer. The predictor predicts the next mobility and location of a MO. The quantizer reduces message traffic. The agent sends quantized messages to a cluster manager. The cluster manager integrates quantized messages from all agents and transmits the integrated quantized message to a central manager. The central manager sends messages from several cluster managers to a location server for storing the location data of MOs.

Figure 11 shows the process of the MPQ-CDM, which consists of six steps: (1) prediction of a MO's location; (2) quantization; (3) acquisition of a MO's real location; (4) message transmission; (5) message reception; (6) location update.

(1) Prediction of a MO's location. This step predicts the location of a MO using the SOES model and calculates a moving distance using $SQRT[(x_p - x_1)^2 + (y_p - y_1)^2]$ where x_p and y_p are the predicted coordinates of a MO, and x_1 and y_1 are the last reported coordinates of a MO.

(2) Quantization. This step compares the threshold with the moving distance of a MO. If the moving distance of a MO is longer than the size of the threshold, the next step is executed. However, if the moving distance is shorter than the size of the threshold, the next step is not executed.

(3) Acquisition of the real location of a MO. If a moving distance is longer than the size of the threshold, the MOLMS acquires the real location of a MO.

(4) Message transmission. In this step, an agent of the MOLMS transmits the current location of a MO to the cluster manager.

(5) Message reception. In this step, the cluster manager receives a location message from an agent.

(6) Location update. If the cluster manager receives a location message, then the location of a MO is updated. However, if the cluster manager does not receive a location message, the current location of a

MO is the last reported location, and the MO's location is not updated.

Each MO moves to its next location from its current location. The MPQ-CDM scheme predicts the mobility of a MO using the SOES prediction model [14], decides the next location of a MO, and quantizes information about the next location. The mobility prediction in the MPQ-CDM scheme depends on the historical movement data of a MO. The quantization of the MPQ-CDM scheme is completed when the next location is crosses a previously assigned boundary (i.e. threshold). Meanwhile, the quantization is not operated when the next location has not crossed a boundary.

Here, the mobility prediction-based quantization of the MPQ-CDM scheme is different from the location quantization of the predictive quantization scheme proposed in Lee and Zeigler [1] and Zeigler et al. [2]. In the predictive quantization scheme, the discrete event system specification (DEVS) predictive integrator [1, 2] based on the DEVS theory [16] predicts the next distance of a MO and quantizes information about the next location in every movement of a MO. The DEVS predictive integrator does not check if the next location has crossed a previously assigned boundary. The DEVS predictive integrator always predicts the next distance with an assigned exact boundary and quantizes information about the next location.

4. Performance Analysis

We analyze the location accuracy and the number of transmitted messages that are required among a cluster manager and several agents for a performance evaluation of the FQ-CDM, AQ-CDM, and MPQ-CDM. For this analysis, we assume that every MO moves every second and that the number of MOs does not change during the message communication time.

Table 1 shows an analysis of the number of required transmission messages of all MOs. N is the number of MOs and T represents a message communication time. R is the rate of MO movement within a threshold limit in the quantization-based CDM schemes. R has a value of $0 \leq R \leq 1$. Thus, R_{FQ}, R_{AQ}, and R_{MPQ} are the rates of the FQ-CDM, AQ-CDM, and MPQ-CDM, respectively. The differences between these rates are $R_{AQ} \geq R_{FQ} = R_{MPQ}$. The FQ-CDM and MPQ-CDM apply a unique threshold to all MOs, and thus R_{FQ} is equal to R_{MPQ}. However, the AQ-CDM constructs MO clusters by classification according to mobility patterns of MOs and uses various thresholds that suit each cluster. Thus, R_{AQ} is greater than R_{FQ} and R_{MPQ}. P is the prediction accuracy rate of the prediction model in the MPQ-CDM and has a value of $0 \leq P < 1$. The value of P cannot be 1 because the MPQ-CDM executes a quantization operation. The non-quantization system is a real-time system and transmits location information when a MO moves.

Table 1. Analysis of the number of required transmission messages of all MOs. N is the number of MOs, R_{FQ} is the rate of MO movement within a threshold in the FQ-CDM ($0 \leq R_{FQ} \leq 1$), R_{AQ} is the rate of MO movement within a threshold in the AQ-CDM ($0 \leq R_{AQ} \leq 1$), R_{MPQ} is the rate of MO movement within a threshold in the MPQ-CDM ($0 \leq R_{MPQ} \leq 1$), $R_{AQ} \geq R_{FQ} = R_{MPQ}$, P is the prediction accuracy rate of the MPQ-CDM ($0 \leq P < 1$), and T is the message communication time

	Number of required transmission messages
Non-quantization	$N \times T$
FQ-CDM	$N \times (1 - R_{FQ}) \times T$
MPQ-CDM	$N \times (1 - R_{MPQ}) \times P \times T$
AQ-CDM	$N \times (1 - R_{AQ}) \times T$

The number of required transmission messages of a non-quantization system is $N \times T$, because we assume that all MOs move once in a second. In the FQ-CDM, the number of required transmission messages is $N \times (1 - R_{FQ}) \times T$, because the agents of the FQ-CDM do not transmit messages when a MO moves within a threshold. Therefore, the number of required transmission message of the FQ-CDM is $(1 - R_{FQ})$ times less than that of the non-quantization. In the MPQ-CDM, the number of required transmission messages is $N \times (1 - R_{MPQ}) \times P \times T$, because the MPQ-CDM depends on prediction results and transmits location information when the prediction results expect that a MO will move a longer distance than a threshold. The number of required transmission messages of the MPQ-CDM is $(1 - R_{MPQ}) \times P$ times less than that of the non-quantization and is P times less than that of the FQ-CDM. P is the prediction accuracy of the next location of a MO. The numbers of required transmission messages of the MPQ-CDM and FQ-CDM are the same when their prediction accuracy rates are 100%. However, the number of required transmission messages of the MPQ-CDM is less than that of the FQ-CDM, because the prediction accuracy rate of the prediction model in the MPQ-CDM is not perfect. In the AQ-CDM, the number of required transmission messages is $N \times (1 - R_{AQ}) \times T$, because the agents of the AQ-CDM do not transmit messages when a MO moves within a threshold. The number of required transmission messages of the AQ-CDM is also $(1 - R_{AQ})$ times less than that of the non-quantization. The AQ-CDM transmits fewer messages than the FQ-CDM, because R_{AQ} is larger than R_{FQ}. Comparing the AQ-CDM with the MPQ-CDM, equation (3) represents when the AQ-CDM transmits fewer messages than the MPQ-CDM:

$$\frac{N \times (1 - R_{AQ}) \times T}{N \times (1 - R_{MPQ}) \times P \times T} < 1. \tag{3}$$

Equation (3) is equal to equations (4)–(7):

$$\frac{(1 - R_{AQ})}{(1 - R_{MPQ}) \times P} < 1 \tag{4}$$

$$(1 - R_{AQ}) < (1 - R_{MPQ}) \times P \quad (5)$$

$$R_{AQ} > P \times (R_{MPQ} - 1) + 1 \quad (6)$$

$$R_{AQ} > P \times R_{MPQ} + (1 - P). \quad (7)$$

Therefore, the AQ-CDM transmits fewer messages than the MPQ-CDM when $R_{AQ} > P \times R_{MPQ} + (1 - P)$. However, if R_{AQ} is less than $P \times R_{MPQ} + (1 - P)$, the AQ-CDM transmits more messages than the MPQ-CDM.

Table 2 shows an analysis of the location accuracy rate of MOs. The location accuracy rate is represented with a probability-related variable and shows how many accurate locations a location server knows. R is the MO movement rate within a threshold limit in the quantization-based CDM schemes. R has a value of $0 \leq R \leq 1$. Thus, R_{FQ} is the rate of the FQ-CDM and R_{AQ} is the rate of the AQ-CDM. Differences between these rates are $R_{AQ} \geq R_{FQ}$. P is the prediction accuracy rate of the prediction model of the MPQ-CDM and has a value of $0 \leq P < 1$, and C is constant. We assume that a location accuracy rate of the non-quantization is 100%. The location accuracy rate of the FQ-CDM is $(1 - R_{FQ}) \times 100\% + C$. The FQ-CDM is able to acquire the exact location of a MO when it moves a longer distance than a threshold. When a MO moves within a threshold, the FQ-CDM is not able to guarantee the location accuracy rate of a MO. Constant C is the location accuracy rate of a MO when it moves within a threshold. If 50% of MOs move within a threshold, then the location accuracy rate of the FQ-CDM is 50% + C. The location accuracy rate of the AQ-CDM is $(1 - R_{AQ}) \times 100\% + C$. The AQ-CDM has a lower location accuracy rate than the FQ-CDM, because R_{AQ} is larger than R_{FQ}. In the MPQ-CDM, the location accuracy rate is $P \times 100\%$. This results from $[(1 - R_{MPQ}) \times P + R_{MPQ} \times P] \times 100\%$. $(1 - R_{MPQ}) \times P$ is the location accuracy rate when a MO moves over a threshold and $R_{MPQ} \times P$ is the location accuracy rate when a MO moves within a threshold. Thus, the location accuracy rate of the MPQ-CDM depends on the prediction accuracy rate of the prediction model.

5. Experiments and Results

In order to evaluate system performance, we experiment and compare the three quantization-based CDM schemes, the FQ-CDM, the AQ-CDM, and the MPQ-CDM. We design and develop the HLA-based MOLMS in a distributed computing environment and apply the three quantization-based CDM schemes to the HLA-based MOLMS. We use a maximum of 10,000 MOs with a maximum speed of 10 m s^{-1} in an area of 1 km^2, and we monitor the movements of the MOs for 6000 s (100 min). As described in Section 3.1, a MO has three movement states: linear movement, random, and stop. Thus, in this experiment each MO has its initial movement state, initial coordinates, direction, and range of moving velocity under 10 m s^{-1}.

Table 2. Analysis of the location accuracy rate of MOs. R_{FQ} is the rate of a MO movement within a threshold in the FQ-CDM ($0 \leq R_{FQ} \leq 1$), R_{AQ} is the rate of a MO movement within a threshold in the AQ-CDM ($0 \leq R_{AQ} \leq 1$), $R_{AQ} \geq R_{FQ}$, P is the prediction accuracy rate of the MPQ-CDM ($0 \leq P < 1$), and C is constant

	Location accuracy rate of MOs
Non-quantization	100%
FQ-CDM	$(1 - R_{FQ}) \times 100\% + C$
AQ-CDM	$(1 - R_{AQ}) \times 100\% + C$
MPQ-CDM	$P \times 100\%$

Each MO moves with its movement state, then it randomly changes its movement state to other states as time goes by. We conduct three types of experiments. The first experiment is to measure a prediction accuracy rate of the SOES prediction model with the MPQ-CDM, as mentioned in Section 3.4. In the first experiment, we decide on the value of the smoothing constant (α), which greatly affects the prediction accuracy rate of the SOES model, and we predict the next coordinates of a MO using this value of the smoothing constant. In the second and third experiments, we compare the system performances of the FQ-CDM, AQ-CDM, and MPQ-CDM. In the second experiment we measure the number of transmitted messages when the three quantization-based CDM schemes use high and low thresholds. In the third experiment we measure the average location accuracy rate of all MOs according to message traffic reduction.

5.1 Value of the Smoothing Constant for the Prediction Model

We decide on the value of the smoothing constant of the SOES, which is the prediction model of the MPQ-CDM. Theoretically, the value of the smoothing constant is between 0 and 1. Thus, we compare the prediction accuracy rate by changing the smoothing constant value from 0.1 to 0.9 in steps of 0.1. Table 3 represents the prediction accuracy rate according to the variations of the value of the smoothing constant. The highest prediction accuracy rate is 88.79% when the value of the smoothing constant is 0.4. However, we use 0.7 for the value of the smoothing constant in the following experiments. This is because the average of the total prediction accuracy rate is 87.34% and the closest value of the smoothing constant to the average prediction accuracy rate is 0.7.

Figure 12 represents the variation of the prediction accuracy rate of the MPQ-CDM during the initial 200 s. From the start time in Figure 12, the prediction accuracy rate is not precise as the data for MOs are not sufficient. However, we know that the prediction accuracy rate increases rapidly and becomes stable as time goes by. The prediction model of the MPQ-CDM has an average prediction accuracy rate of 87.34% over 6000 s. Therefore,

Table 3. Prediction accuracy rate of the SOES prediction model according to the variations in the value of the smoothing constant

Smoothing constant	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Average
Prediction accuracy rate (%)	85.74	87.71	88.53	88.76	88.49	87.82	87.45	86.41	85.13	87.34

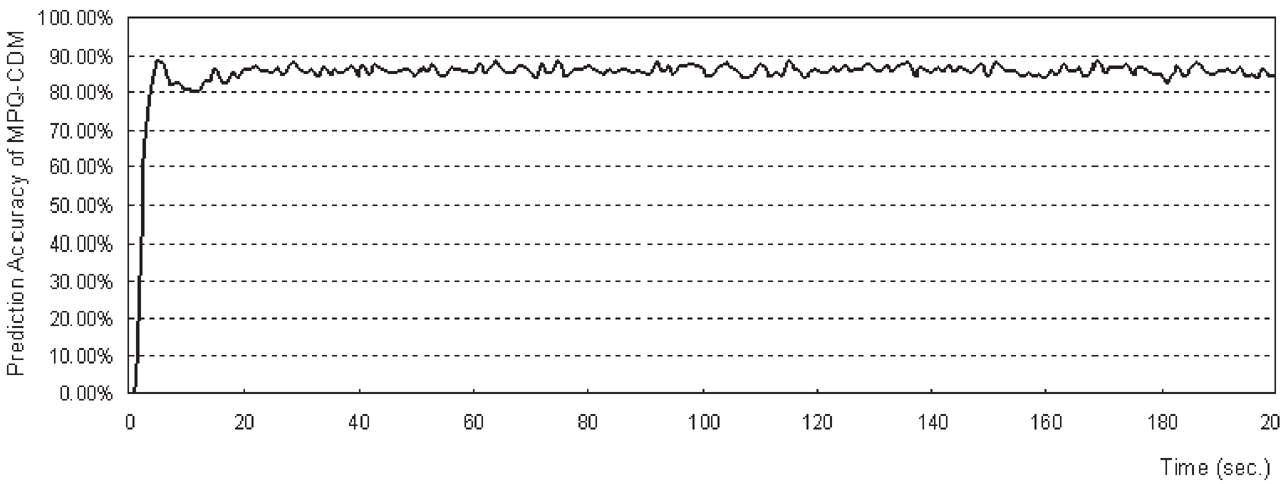


Figure 12. Variation of the prediction accuracy rate of the MPQ-CDM

Table 4. Meaning of the MAPE

MAPE	Meaning
0% $\leq$ MAPE $<$ 10%	Very accurate prediction
10% $\leq$ MAPE $<$ 20%	Comparative accurate prediction
20% $\leq$ MAPE $<$ 50%	Comparative rational prediction
MAPE $\geq$ 50%	Non-accurate prediction

we can estimate the location accuracy rate of all MOs in the MPQ-CDM by using the analysis of Table 2. The location accuracy rate of all MOs in the MPQ-CDM is expected to be close to 87% of the non-quantization system.

As described above, the location accuracy rate of all MOs in the SOES prediction model is close to 87%. Here, we need to know whether the location accuracy rate of all MOs is reliably predicted. Generally, the reliability of prediction is evaluated with the mean absolute percentage error (MAPE) [17]. The MAPE is expressed as

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{X_t - F_t}{X_t}\right| \times 100(\%) \qquad (8)$$

where X_t is the measured value, F_t is the predicted value, and n is the number of time intervals for measurement. The meaning of the reliability of prediction with the MAPE depends on the range of the MAPE, as shown in Table 4.

Figure 13 expresses the MAPE of the SOES prediction model during the initial 200 s. The MAPE has a value below 10%. According to Table 4, this MAPE means "very accurate prediction". However, this result is only for the initial 200 s. Thus, the MAPE should be below 10% for the 6000 s of the entire experimental time. In order to predict a change of the MAPE, we use linear regression analysis [18] and make a mathematical model that expresses a change of the MAPE (Figure 13). The linear regression analysis is expressed as (see also Figure 14) [18]

$$y = -0.007x + 9.917. \qquad (9)$$

According to equation (9), we can expect the MAPE to be below 10%. Therefore, we know that the prediction of the SOES model is reliable.

5.2 Comparison of the Number of Transmitted Messages

In this experiment, we compare four types of quantization-based CDM scheme: non-quantization, FQ-CDM, AQ-CDM, and MPQ-CDM. We measure the number of transmitted messages by using high and low thresholds. The non-quantization system does not perform any quantization. In the MOLMS with the non-quantization system, an agent transmits a location message when a MO moves to the next location. Thus, an agent of the MOLMS with the non-quantization system transmits location messages to a

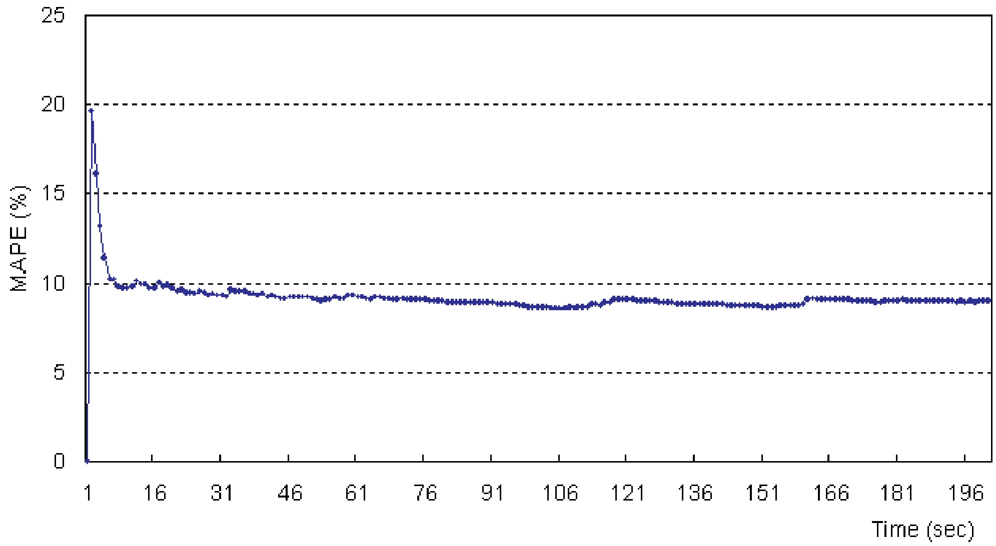


Figure 13. MAPE of the SOES prediction model.

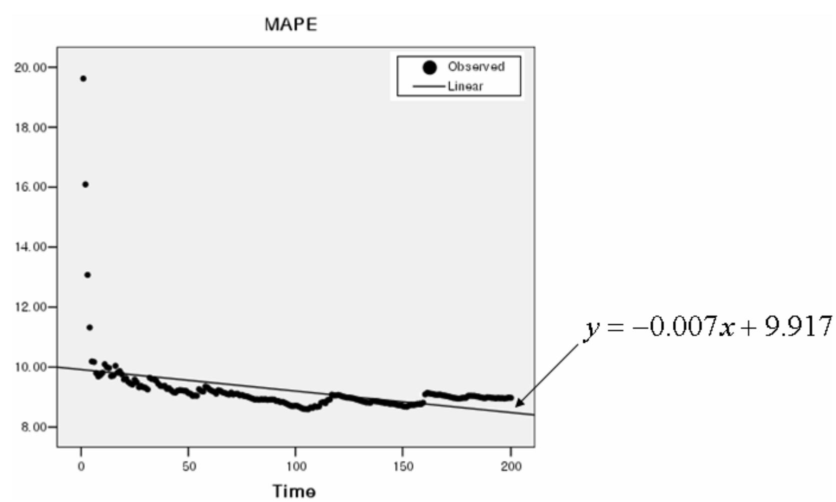


Figure 14. Linear regression analysis of the MAPE

cluster manager every second (i.e. every discrete event). Figures 15 and 16 represent the number of transmitted messages with low and high threshold, respectively. The number of transmitted messages of the AQ-CDM is less than that of the non-quantization system, FQ-CDM, and MPQ-CDM with both high and low thresholds. For both low and high thresholds, the non-quantization system transmitted 60,000,000 messages. For a low threshold, the AQ-CDM transmitted 43,468,407 messages over 6000 s. The MPQ-CDM transmitted 45,755,600 messages and the FQ-CDM transmitted 47,641,100. The AQ-CDM shows 4.9% more reduction than the MPQ-CDM and 8.7% more reduction than the FQ-CDM. Also, the AQ-CDM shows 27.55% reduction compared with the non-quantization. For a high threshold, the AQ-CDM transmitted 28,662,752 messages over 6000 s. The MPQ-CDM

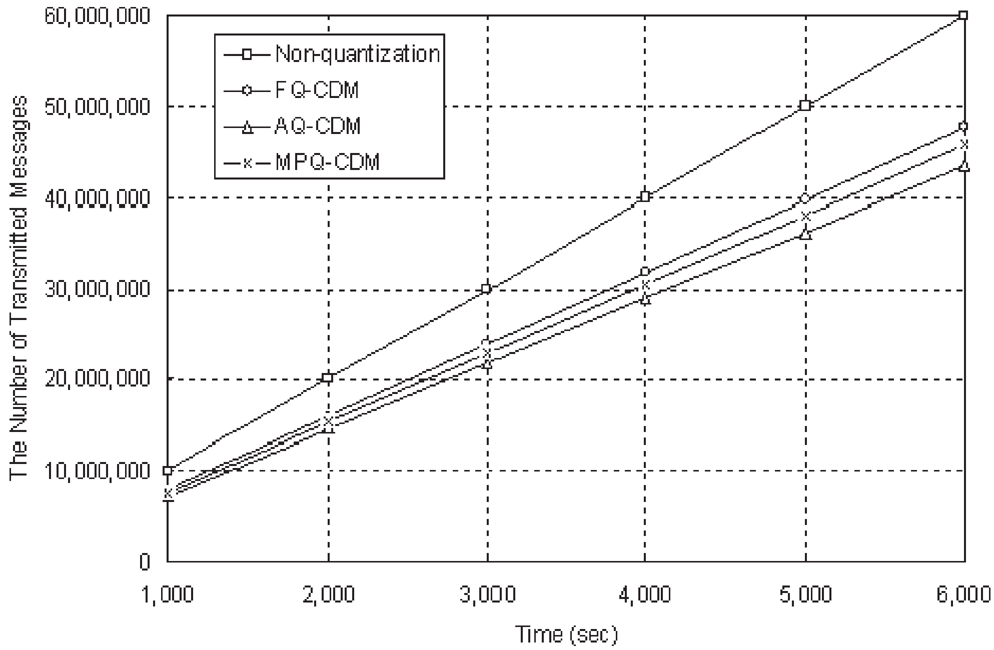


Figure 15. Comparison of the number of transmitted messages with a low threshold for non-quantization, FQ-CDM, AQ-CDM, and MPQ-CDM

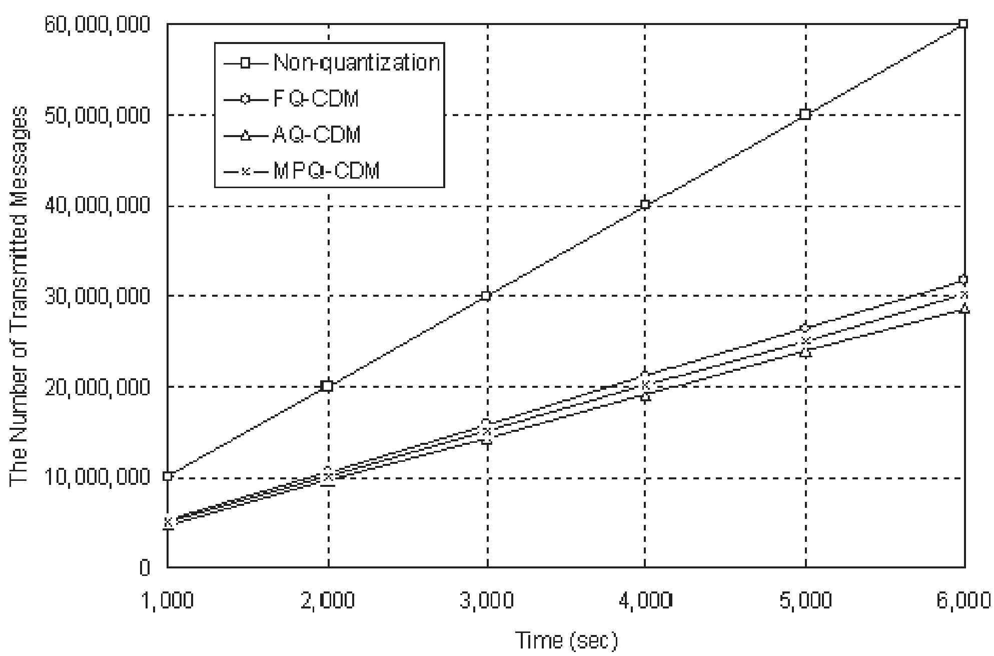


Figure 16. Comparison of the number of transmitted messages with a high threshold for non-quantization, FQ-CDM, AQ-CDM, and MPQ-CDM

Table 5. Comparison of the RMSE of the location errors of the FQ-CDM, AQ-CDM, and MPQ-CDM

Time (s)	Low threshold			High threshold		
	FQ-CDM	AQ-CDM	MPQ-CDM	FQ-CDM	AQ-CDM	MPQ-CDM
1000	197.56	176.96	65.85	604.58	569.04	194.40
2000	395.28	401.46	131.76	1159.09	1118.99	388.96
3000	593.10	624.36	197.70	1757.89	1872.29	583.61
4000	790.88	848.44	263.63	2186.81	2535.72	778.23
5000	988.56	1068.38	329.52	3015.50	2949.58	972.74
6000	1186.38	1289.14	395.46	3408.80	3933.96	1167.40
Sum	4151.76	4408.74	1383.92	12132.68	12979.58	4085.33

and FQ-CDM transmitted 30,170,700 and 31,615,600 messages, respectively. The AQ-CDM shows 4.9% more reduction than the MPQ-CDM and 9.3% more than the FQ-CDM. The total reduction rate of the AQ-CDM with a high threshold is 52.22%. With the reduction of transmitted messages, the AQ-CDM is a more effective CDM scheme than the MPQ-CDM and FQ-CDM for both high and low thresholds. The AQ-CDM transmits fewer messages than the MPQ-CDM for the following reasons. First, the AQ-CDM creates MO clusters through the mobility patterns of MOs and uses a suitable size of threshold. Thus, R_{AQ} of Table 1 is greater than R_{MPQ}. Secondly, the average prediction accurate rate of the MPQ-CDM is 87.34% and $(1 - P)$ of equation (7) is close to 0.13. The fact that R_{AQ} is greater than $0.87 \times R_{\mathrm{MPQ}} + 0.13$ is reasonable. Therefore, concerning the reduction of transmitted messages, the AQ-CDM is more effective than the MPQ-CDM. However, the location error rate is generally increased as the number of transmitted messages is decreased. Thus, in the following experiment, we measure the location error rates of the three quantization-based CDM schemes in accordance with message reduction.

5.3 Comparison of Location Error

In this experiment, we measure the location error rate, which occurs as a result of quantization operations of the FQ-CDM, AQ-CDM, and MPQ-CDM schemes. This experiment to measure the location error rate is conducted for both high and low thresholds. We use the root mean square error (RMSE) [19] to measure the location error rate. The RMSE is generally used in the field of statistics and denotes the average difference between real and estimated values [19]. We use the RMSE to measure the degrees of the location error rate. The RMSE is expressed as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(RCoord - ECoord)^2} \qquad (10)$$

where *RCoord* are the real coordinates of MOs, *ECoord* are the estimated coordinates of MOs, and n is the total number of locations of MOs. Table 5 shows the RMSE of location errors of the three quantization-based CDM schemes. We can see that the MPQ-CDM has a smaller RMSE than the other quantization-based CDM schemes for both low and high thresholds. The MPQ-CDM is more accurate when the locations of MOs are more exactly estimated. The MPQ-CDM uses the SOES prediction model and is able to predict the locations of MOs even when message reduction operations are performed.

The location accuracy rate is 100% when the location server of the MOLMS knows the exact location of a MO. Table 6 represents the probabilistic accuracy rate of a MO's location in the three quantization CDM schemes (i.e. the FQ-CDM, AQ-CDM, and MPQ-CDM). Table 6 shows two types of location accuracy: the location accuracy rate in quantization and the total location accuracy rate. The location accuracy rate in quantization means that the location server of the MOLMS has the location accuracy rate when each CDM scheme operates its quantization process. The total location accuracy rate is the average rate of two location accuracy rates in quantization and non-quantization. The location accuracy rate in quantization of the FQ-CDM is 56.10% with a low threshold and 51.86% with a high threshold. The AQ-CDM has 57.52% and 53.15% of the location accuracy rate in quantization with low and high thresholds, respectively. For the MPQ-CDM, the location accuracy rate in quantization is 81.43% and 75.25% with low and high thresholds, respectively. The FQ-CDM shows a total location accuracy rate of 88.75% and 77.19% with low and high thresholds, respectively. The AQ-CDM shows a total location accuracy rate of 86.24% and 76.35% with low and high thresholds, respectively. The total location accuracy rate of the MPQ-CDM is 96.14% and 89.26% for low and high thresholds, respectively.

For the AQ-CDM, the location accuracy rates in quantization for both low and high thresholds are the higher than those of the FQ-CDM. However, for the total location accuracy rates, the AQ-CDM shows lower rates than those

Table 6. Comparison of the location accuracy rates of the FQ-CDM, AQ-CDM, and MPQ-CDM for low and high thresholds

	Low threshold			High threshold		
	FQ-CDM	AQ-CDM	MPQ-CDM	FQ-CDM	AQ-CDM	MPQ-CDM
Location accuracy rate in quantization	56.10%	57.52%	81.43%	51.86%	53.15%	75.25%
Total location accuracy rate	88.75%	86.24%	96.14%	77.19%	76.35%	89.26%

of the FQ-CDM, because the AQ-CDM uses fewer communication messages than the FQ-CDM by using suitable clusters for MOs. Thus, the AQ-CDM eventually results in a less accurate total location accuracy rate than the FQ-CDM.

For the MPQ-CDM, the location accuracy rates in quantization are 81.43% and 75.25% for low and high thresholds, respectively, and the total location accuracy rates are 96.14% and 89.26% for low and high thresholds, respectively. Thus, the MPQ-CDM shows the best performance for the two accuracy rates (i.e. the location accuracy rate in quantization and the total location accuracy rate), because the MPQ-CDM can estimate the next location of a MO by using a reliable prediction model. Consequently, this experiment to measure the two location accuracy rates shows that the MPQ-CDM has higher location accuracy rates than the other quantization-based CDM schemes.

6. Conclusion

The CDM scheme is an important issue in complex and large-scale distributed systems, which should reduce communication traffic effectively and manage a fault-tolerant system. In this paper, we propose and compare quantization-based CDM schemes, such as the FQ-CDM, AQ-CDM, and MPQ-CDM, which are applicable in a high performance distributed computing environment. The FQ-CDM applies the basic theory of a quantized system to a CDM scheme in distributed computing. The AQ-CDM and the MPQ-CDM are improved forms of the FQ-CDM. The AQ-CDM uses classification and clustering algorithms to construct MO clusters using the movement characteristics of MOs. The classification and clustering algorithms enable the AQ-CDM to use various and suitable thresholds. The MPQ-CDM uses a prediction model of statistics and predicts the next mobility and location. The MPQ-CDM can reduce communication messages using quantization only if a MO is expected to move more than a threshold limit.

In this paper, the MOLMS is designed and implemented to apply the three quantization-based CDM schemes to a real distributed computing. Each of quantization-based CDM schemes is applied to the MOLMS. The performance evaluation of the three quantization-based CDM schemes is conducted through analysis and experiments on the MOLMS. In an experiment, the optimal value of the smoothing constant, which is a critical variable in prediction with the SOES model, is decided. The number of transmitted messages and the location accuracy rates are measured. The AQ-CDM is an effective solution for reducing communication traffic. The MPQ-CDM is more effective at providing better location accuracy, and thus it can be a good solution when location accuracy is very important.

7. Acknowledgment

This work was supported by a research grant from Inha University.

8. References

[1] Lee, J. S., and Zeigler, B. P. 2002. Space-based communication data management in scalable distributed simulation. *Journal of Parallel and Distributed Computing* 62:336–65.

[2] Zeigler, B. P., Cho, H. J., Kim, J. G., Sarjoughian, H. S., and Lee, J. S. 2002. Quantization-based filtering in distributed discrete event simulation. *Journal of Parallel and Distributed Computing* 62:1629–74.

[3] Defense Modeling and Simulation Office (DMSO). 1998. *High Level Architecture Run-Time Infrastructure Programmer's Guide 1.3*. 1998. Version 3. Alexandria, VA, USA: DMSO.

[4] Boukerche, A., and Dzermajko, C. 2003. Dynamic grid-based vs. region-based data distribution management strategies for large-scale distributed systems. In *Proceedings of the 17th IEEE International Parallel and Distributed Processing Symposium (IPDPS'03)*, April, Nice, France.

[5] Boukerche, A., and Roy, A. J. 2002. Dynamic grid-based approach to data distribution management. *Journal of Parallel and Distributed Computing* 62:366–92.

[6] Tan, G., Ayani, R., Zhang, Y., and Moradi, F. 2000. Grid-based data management in distributed simulation. In *Proceedings of the 33rd Annual Simulation Symposium (SS 2000)*, April, Washington, DC, USA.

[7] Boukerche, A., and Dzermajko, C. 2004. Dynamic grid-based vs. region-based data distribution management strategies in multi-resolution large-scale distributed systems. In *Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS'04)*, April, Santa Fe, NM, USA.

[8] Boukerche, A., and Kaiyuan, L. 2005. Optimized dynamic grid-based DDM protocol for large-scale distributed simulation systems. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, April, Denver, CO, USA.

[9] Song, M. B., Ryu, J. H., Lee, S. K., and Hwang, C. S. 2003. Considering mobility patterns in moving objects database. In *Proceedings of the 32nd IEEE International Conference on Parallel Processing Symposium (ICPP'03)*, October, Kaohsiung, Taiwan.

[10] Yang, M. H., Chen, L. W., Sheu, J. P., and Tseng, Y. C. 2001. A traveling salesman mobility model and its location tracking in PCS networks. In *Proceedings of the 21st IEEE International Conference on Distributed Computing System (ICDCS 01)*, April, Phoenix, AZ, USA.

[11] Saglio, J.-M., and Moreira, J. 2001. Oporto: a realistic scenario generator for moving objects. *GeoInformatica* 5(1):71–93.
[12] DMSO, United States Department of Defense, https://www.dmso.mil/public/public/.
[13] Theodoridis, S., and Koutroumbas, K. 2006. *Pattern Recognition,* 3rd edn. New York: Academic Press.
[14] McClave, J., Benson, P., and Sincich, T. 2004. *Statistics for Business and Economics*, 9th edn. Englewood Cliffs, NJ: Prentice Hall.
[15] Kreyszig, E. 2005. *Advanced Engineering Mathematics*, 9th edn. New York: Wiley.
[16] Zeigler, B. P., Kim, T. G., and Praehofer, H. 2000. *Theory of Modeling and Simulation*, 2nd edn. New York: Academic Press.
[17] Makridakis, S. G., Wheelwright, S. C., and Hyndman, R. J. 1997. *Forecasting*: *Methods and Applications*, 3rd edn. New York: Wiley.
[18] Mendenhall, W., and Sincich, T. L. 2006. *A Second Course in Statistics*: *Regression Analysis*, 6th edn. Englewood Cliffs, NJ: Prentice Hall.
[19] Ghilani, C. D., and Wolf, P. R. 2006. *Adjustment Computations*: *Spatial Data Analysis*, 4th edn. New York: Wiley.

In Kee Kim *is a graduate student at the School of Computer Science and Information Engineering, Inha University, Incheon, South Korea.*

Sung Ho Jang *is a graduate student at the School of Computer Science and Information Engineering, Inha University, Incheon, South Korea.*

Jong Sik Lee *is an associate professor at the School of Computer Science and Information Engineering, Inha University, Incheon, South Korea.*