A Scalable Data Distribution Management Approach

Shih-Hsiang Lo and Yeh-Ching Chung
Department of Computer Science, National Tsing Hua University, Taiwan
albert@sslab.cs.nthu.edu.tw and ychung@cs.nthu.edu.tw

Keywords: Data Distribution Management, High Level Architecture, Run-time Infrastructure

Abstract

Using Data Distribution Management (DDM), federates can send/receive interest data properly instead of broadcasting data over entire network. For large-scale HLA simulations with many simulation entities or federates, the use of DDM is essential to support such simulations. However, the scalability issue of DDM is rarely raised in the literature. Thus we present a scalable DDM approach, DHT-DDM, using Content-Addressable Networks (CANs) [1] based on Distributed Hash Table (DHT). Region declaration, region matching, and network connection mechanisms for DDM are designed considering scalability. The declaration of regions is done deterministically without depending on global information, the load of region matching is distributed and shared among federates, and the connections of federates is built upon application-level multicast.

1. INTRODUCTION

The High Level Architecture (HLA) [2] is a general-purpose framework for distributed simulation and modeling. Based on this distributed simulation framework, various HLA compliant simulators or models (termed 'federates') can be integrated into a simulation (termed 'a federation'). The HLA interface specification [3] specifies how a federate interacts with other federates via run-time infrastructure (RTI). The RTI is a middleware that provides management services for federates. Data Distribution Management, one optional service in RTI, is used to reduce the irrelevant transmission between federates.

By the DDM service, a federate can declare its interest regions and then the connections between interest federates are built. In general, a DDM process consists of three phases: region declaration, region matching, and network connection phases. In the region declaration phase, interest regions (i.e., update and subscription regions) of federates (i.e., publishers and subscribers) are sent to region matching processes. In the region matching phase, update and subscription regions of federates are matched in order to find overlapping results by region matching processes. In the network connection phase, network connections between publishers and subscribers are connected/disconnected according to the overlapping results obtained from the region matching processes. After a DDM process, federates can send the data directly to the corresponding interest federates.

As the number of simulation entities or federates in a simulation increases significantly, a scalable DDM is indispensable for supporting the execution of such a simulation. The proposed DDM approaches are the region-based approach [4-5], the grid-based approach [6], the hybrid-based approach [7-8], the partition-based approach [9-10], the agent-based approach [11-13] and the sort-based approach [14-17]. However, the scalability issue of DDM is rarely explored. For the region-based approach known as brute-force approach, its region matching algorithm limits the scalability of DDM. For the grid-based approach, the resource of multicast groups used is scarce and limited and the performance of DDM is highly related to the chosen grid cell size. For the hybrid-based approach, it has the same problems as the grid-based approach. For the agent-based approach, if a simulation involves many federates, there is no easy mean to distribute agents, which are used to carry the regions of subscribers, to the proper publishers. For the sort-based approach, it largely pays attention to the region matching algorithm. For the partition-based approach, federates are required to know the overall partition result when declaring regions.

To address the scalability issue of DDM, every part of DDM (i.e., the region declaration, the region matching and the network connection phases) needs to be considered:

Region declaration phase: regions can be transmitted to the corresponding region matching process(es) deterministically without depending on global information about region matching processes.
Region matching phase: the load of region matching can be distributed and shared among region matching processes if some region matching processes are overloaded with work.
Network connection phase: connections between federates can be established at application-level to keep from binding specified algorithms and multicast techniques, e.g., IP multicast.

We therefore propose a DDM approach, DHT-DDM, to deal with the scalability issue of DDM. The proposed approach uses Content-Addressable Networks (CANs) [1] based on Distributed Hash Table (DHT) to make federates connect in a structured way. Each federate joins CANs as a CAN node. A CAN node owns a distinct space, called a zone, and performs region matching for the regions which belongs in its own zone. In this way, regions of federates can be transmitted to some nodes using the CAN routing scheme, the load of region matching can be distributed and shared among the nodes by adjusting routing tables of nodes, and the network connections between publishers and subscribers are connected/disconnected by joining/leaving the application-level multicast groups.

The organization of this paper is as follows. In Section 2, we briefly discuss the DDM approaches reported in the literature. The design of DHT-DDM is shown in Section 3. The details of DHT-DDM are presented in Section 4. We conclude the paper in Section 5.

2. RELATED WORK

2.1. Region-Based Approach

When the range of an update (subscription) region is modified, this approach compares this modified region with all subscription (update) regions to find overlapping results [4-5]. The time complexity of the region-based approach is quadratic with the number of regions. The idea is straightforward but in large-spatial simulations this approach could take much time to compare unrelated regions. On the other hand, it can achieve high performance when all regions are highly overlapped [18-19].

2.2. Grid-Based Approach

In [6], the authors proposed a method to reduce the region matching cost of the region-based approach. This approach divides an *N*-dimensional space into grid cells and maps all regions to these grid cells. Each grid cell has a multicast group. When at least one update region and one subscription region are mapped to the same grid cell, both regions are presumed to overlap with each other and the corresponding federates join the same multicast group. The computation time of this mechanism is much less than that of the region-based approach. However, the number of multicast groups available depends on the chosen network infrastructure and also limits the number of grid cells created. As a result, the grid-based approach address resource allocation and transmission control issues of DDM. The DDM approaches in [20-23] address the use of multicast groups. The transmission control of messages generated for the grid-based DDM has been studied in [24-26].

2.3. Hybrid-Based Approach

A combination of the region-based and the grid-based approaches has been devised in [7-8]. The region matching of the hybrid-based approach consists of two phases. In the first phase, an *N*-dimensional space is divided into a set of grid cells. In the second phase, when an update/subscription region is modified, the update/subscription region is first sent to a centralized cell manager or a couple of cell managers. In [7], a centralized cell manager (i.e., DDM coordinator) is used to manage all grid cells. Conversely, in [8], each cell manager (i.e., local RTI component (LRC)) is used to exclusively manage some grid cell(s). Then the cell manager determines overlapping results by applying the region-based approach to the proper grid cells. This approach can remove the irrelevant messages generated and reduce computational overhead. However, an inappropriate grid cell size greatly affects the performance of this approach. The large grid cell size could lead to unnecessary computation on comparing unrelated regions. Conversely, the small grid cell size could lead to redundant computation on calculating overlapping results for update and subscription regions among different grid cells.

2.4. Agent-Based Approach

In [11-12], the authors used mobile agent technique to carry out data filtering for subscribers. When a subscriber claims a set of subscription regions, a set of mobile agents created for these subscription regions are sent to associate with the corresponding publishers. These agents will eliminate unnecessary data and send the exact data back to the subscriber when those publishers update data. However, this approach has one problem. The problem is how to distribute many mobile agents to publishers in a large-scale simulation.

2.5. Sort-Based Approach

Several region matching approaches using sorting technique have been proposed in [15-17]. In [16], the end points (i.e., the upper bounds and the lower bounds) in each dimension of all regions are first sorted and recorded in a sorted list. The sorted list is then scanned to get overlapping results in each dimension. The overall overlapping results can be obtained by merging the overlapping result of each dimension. This approach has good performance because the calculation of overlapping results is performed before the execution of simulation. However, this approach cannot be applied to simulations where regions will be modified at run-time. For this reason, a dynamic sort-based algorithm is presented in [15] to deal with the problem. When a region is modified, the dynamic sort-based algorithm shifts the end points of this region from old positions to new positions and then scans the sorted end points within a dynamic range. The dynamic range is defined in terms of the end points of this region and the maximum interval length of all update

and subscription regions. As a result, the larger region size a simulation has, the more time it spends on scanning the end points of regions.

In [17], the authors proposed a P-Pruning algorithm for DDM. This approach builds a region projection array to store regions according to the end points of a region. By scanning the end points of a region in the region projection array, the overlapping result of a region can be obtained. The principle of this approach is similar to the work in [15-16]. The sorting procedure of this approach is fast (because bucket sort is used to sort the end points of all regions). However, this mechanism is limited to a small-scale simulation.

2.6. Partition-Based Approach

This approach splits an *N*-dimensional space into fixed-size partitions, similar to grid cells in the grid-based and the hybrid-based approaches. At run-time, it re-partitions the *N*-dimensional space to balance the number of regions among the partitions. In [9], the authors proposed a region matching approach that clusters an *N*-dimensional space into varied-size partitions considering region access patterns as well as the location of simulation object. After the clustering, regions and simulation objects are re-distributed to different hosts. As a result, the execution of region matching can be evenly distributed to different hosts and therefore the bottleneck can be avoided. However, few details of the implementation of this DDM approach are given. A simple partition-based approach using quadtree structure in helping dynamic partition was proposed in [10]. The idea is to split a partition into four equal size partitions until there are no more than two regions in a partition. If there are only two regions in a partition, these two regions are compared to find overlaps. Otherwise, no overlapped regions exist in this partition. In both DDM approaches [9-10], when the partition of a space is altered, all federates need to know the new partition result in order to store the regions to the correct partition.

3. DESIGN OVERVIEW

In a military exercise, the space may be large and a great number of regions exist in the space. To compare many regions in a centralized server (i.e., a region matching process) is inefficient. The load of region matching needs to be distributed among several region matching processes. As a result, we make each federate to share the load of region matching. In addition, regions are transmitted and stored somewhere in order to do region matching.

In this paper, we incorporate CANs into the DDM process because CANs provide a structured topology for ease to access and store data in nodes. Briefly, a CAN is an overlay network where all the nodes form a virtual coordinate space. Each node in a CAN owns an individual space (called a zone) and stores (*Key*,*Value*) pairs if the hashing result of a key is within its zone. Each CAN node maintains a routing table, which records its neighbor nodes and their zones. Routing in a CAN is to follow the straight line through the virtual space from a source node to a destination node. Figure 1 shows a simple example of a 2-dimensional CAN with 5 nodes, n_1, n_2, .., and n_5. The size of the virtual coordinate space is [0,1]×[0,1].

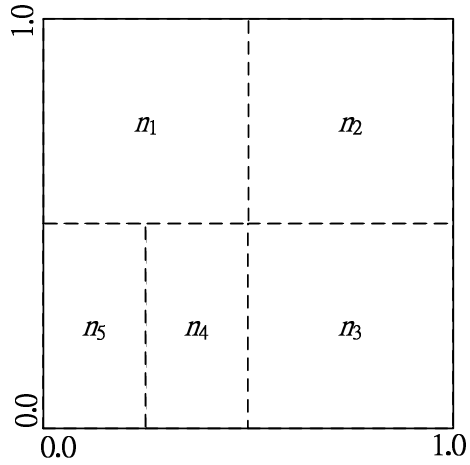

Figure 1. A 2-dimensional CAN with 5 nodes

According to DDM in the HLA, a region is defined as a set of intervals, where an interval is a pair of two values, i.e., the lower and upper bounds. Figure 2 shows an example of regions in a small 2-dimensional space with size [0,100]×[0,100]. There are two update regions, u_1 and u_2, and two subscription regions, s_1 and s_2, in the space. Each region in Figure 2 has two intervals along dimension D_1 and D_2. The overlapping result after region matching indicates that u_1 and u_2 overlap s_1 and s_2, respectively.

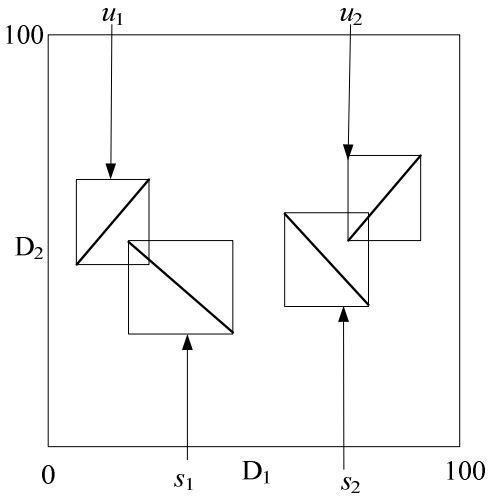

Figure 2. Four regions in a 2-dimensional space
(⊿: update region; ⧅: subscription region)

From Figures 1 and 2, we can observe that a CAN forms a virtual space and regions exist in a space. If we can have a hash function, a region in a space can be mapped on to a virtual coordinate space easily. In our design, an *N*-dimensional space has a corresponding *N*-dimensional virtual space. Then the execution of region declaration, region matching and network connection is based upon CANs. Note that regions with different numbers of intervals could exist in a simulation. These regions will be in different spaces and do not overlap with each other. Different CANs with different virtual coordinate spaces are created to handle these regions. For simplicity, regions are assumed within the same space in the rest of paper.

4. DHT-DDM

In the following, we will describe the proposed DDM approach, DHT-DDM, including the region declaration, the region matching and network connection phases.

4.1. Region Declaration Phase

A federate transmits its interest regions to destination nodes via its neighbor nodes when declaring the regions. A destination node is a node where regions are delivered for performing region matching. This phase consists of two steps.

Step 1: the federate calculate the corresponding points (in a virtual space) of the regions by using a hashing function. Since a region refers to a set of intervals, we thus hash the end point values of a region in each dimension to obtain the virtual points in a CAN space. The hash function used in this paper is defined as $H(X_{Di}) = X_{Di\text{-}CAN} = X_{Di}$ / (the maximum value in D_i), where D_i is the ith dimension and X_{Di} and $X_{Di\text{-}CAN}$ are the points of the ith dimensional in a space and the corresponding virtual space, respectively. In an *N*-dimensional virtual space, the number of virtual points of a region is $2\times N$.

Step 2: the federate transmits the regions to destination nodes using the CAN routing scheme. Briefly, when a node receives a region from its neighbor nodes, if the receiving node is the destination node for the region, the region is stored in this node. If not, the region is routed through the neighbor nodes of the receiving node. Basically, a region is required to be transmitted several times, depending on the number of virtual points the region has. To reduce the number of messages transmitted, one of the virtual points is chosen and then the region is transmitted to the destination node whose zone contains the chosen virtual point. After reaching the destination node, the region will be transmitted to the other destination nodes whose zones contain the other virtual points if necessary. The message aggregation mechanism will reduce the amount of network transmission when all the virtual points of a region belong to the same zone. We now give an example to explain the region declaration phase. In Figure 3, there are 9 federates joining a CAN. When update region u_1 is declared by f_5, the virtual points of u_1 are calculated as shown in Figure 3 and then the region is routed to f_6 via f_1 according to the chosen virtual point. When the f_6 receives the region, f_6 considers the other three virtual points of u_1 and then transmits u_1 to f_2.

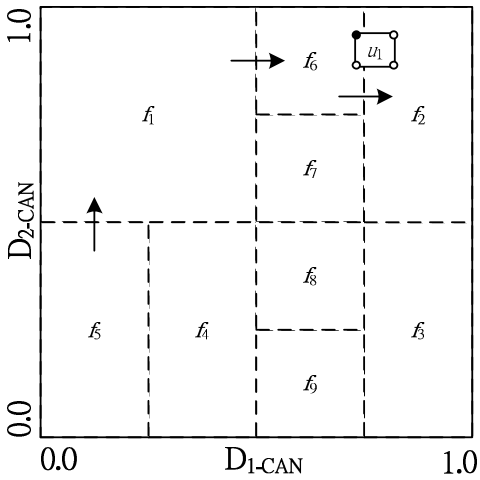

Figure 3. The declaration of region u_1 by f_5 (•: the chosen virtual point of u_1)

4.2. Region Matching Phase

When a node receives a region and performs region matching for the region, the node compares the region with other regions stored in the node. In this phase, we adopt the region matching algorithm of the hybrid-based approach to doing region matching. Initialization, the zone is partitioned into a set of grid cells C_{all} and each grid cell has the same size. When a region is declared (or modified), the region is first mapped to a set of grid cells *C* according to the location of this region. Then, for each grid cell in *C*, the brute-force algorithm is used to determine the overlapping result for this declared (modified) region.

The matches of two regions carried out can be classified into three types: *effective match*, *unnecessary match* and *redundant match*. A match of two regions is an effective match (EM) if two regions are in a given grid cell and they are overlapped. A match of two regions is an unnecessary match (UM) if two regions are in a grid cell and they are not overlapped. A match is a redundant match (RM) if the match of two regions has been performed at other grid cells. UMs and RMs decrease the efficiency of region matching. In the region matching algorithm of the hybrid-based approach, the number of EMs occurred is a constant for the moment. However, the numbers of UMs and RMs occurred are related to the chosen grid cell size. When the chosen grid cell size is larger than the average region size of regions, the number of UMs will increase. Conversely, when the chosen grid cell size is smaller than the average region size of regions, the number of RMs will increase. Both situations result from an inappropriate grid

cell size. To deal with the problem of an inappropriate grid cell size, we use a region matching cost model in [27] to estimate the total number of matches. With the region matching cost model, we can quickly estimate the number of matches under a new grid cell size. If the number of matches can be reduced using a new grid cell, the grid cell size is changed to the new one. If not, the grid cell size is unchanged.

In addition to the problem of an inappropriate grid cell size, in real simulations, simulation entities (or federates) could be interested in data in some hot areas. Consequently, regions are crowded in a zone or several zones. This leads to load imbalance among nodes. To share the load of region matching in a CAN, an overload node first obtains the region matching costs of the overloaded node and its neighbor nodes based on the region matching cost model. If a node can share the load with other neighbor node, the boundary between it and its neighbor node is altered and a partial regions stored in the overloaded node is transferred to the neighbor node.

4.3. Network Connection Phase

After the region matching phase, overlapping results return to the declared federates. If the overlapping result indicates that one subscription region overlaps with one update region, the connection between the corresponding subscriber and publisher is connected by simply making the publisher and the subscriber join a specified CAN, which is used to maintain the multicast group of the publisher, called a MGR CAN. The bootstrap of this MGR CAN is the publisher. Specifically, this MGR CAN is constructed for the publisher and all the subscribers joining this MGR CAN are interested in data from the publisher. If the subscriber has no interest in data from the publisher (i.e., no overlap between their regions), the subscriber just leaves the corresponding MGR CAN. Once the publisher updates its data through other services in RTI (i.e., Object Management), the data floods the MGR CAN for this publisher because all the subscribers in the MGR CAN are interested in data from the publisher. For example, in Figure 4, f_5 is a publisher; the other federates are subscribers. First, f_5 transmits the data to its neighbor federates, f_3, f_4, f_6 and f_9. Then, these neighbor federates transmits the data to their neighbor federates along one direction. The details of the flooding scheme for CANs can refer to [28].

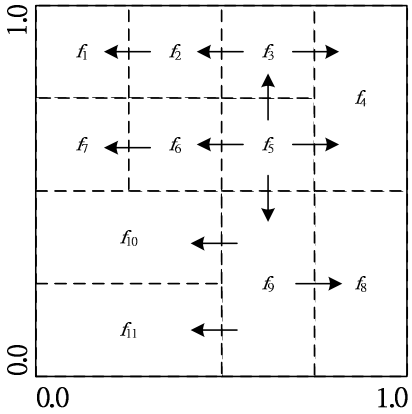

Figure 4. The connection topology of source node f_5 in a MGR CAN

5. CONCLUSIONS AND FUTURE WORK

In this paper, we focus on the scalability issue of DDM and discuss the DDM approaches reported in the literature. To address the issue, we proposed a scalable DDM approach, DHT-DDM, using Content-Addressable Networks based on Distributed Hash Table. The proposed mechanisms have the declaration of regions done deterministically without global information, have the load of region matching balanced among federates, and have the network connections connected/disconnected using application-level multicast. The advantage of DHT-DDM is that the numbers of simulation entities and federates involved in a simulation can increase more, compared with other DDM approaches. The disadvantage of DHT-DDM (using CANs to make federates connected in a logical network) is the long network latency when transmitting messages. However, the long network latency problem can be relieved by considering the physical network topology when a node joining/leaving CANs.

In the future, we will have to make DHT-DDM into practice and evaluate its performance. In order to make good use of DHT-DDM, both the scalability and the performance issues will be considered for various HLA applications. Another issue is how to make an RTI system scale well when other managements such as Time Management are also used in a simulation.

Reference

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," presented at the Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, California, United States, 2001.

[2] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -- Framework and Rules," vol. 1516-2000, ed, 2000.

[3] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -- Interface Specification," vol. 1516.3-2000, ed, 2000.
[4] D. J. Van Hook and J. O. Calvin, "Data Distribution Management in RTI 1.3," in *the 1998 Spring Simulation Interoperability Workshop*, 1998.
[5] D. Wood, "Implementation of DDM in the MAK High Performance RTI," presented at the the 2002 Simulation Interoperability Workshop, 2002.
[6] D. V. Hook, S. Rak, and J. Calvin, "Approaches to RTI Implementation of HLA Data Distribution Management Services," presented at the the 15th Distributed Iinteractive Simulation Workshop, 1996.
[7] G. Tan, Z. Yusong, and R. Ayani, "A hybrid approach to data distribution management," presented at the the Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications, 2000.
[8] A. Boukerche, N. McGraw, C. Dzermajko, and K. Lu, "Grid-Filtered Region-Based Data Distribution Management in Large-Scale Distributed Simulation Systems," presented at the the 38th Annual Simulation Symposium, 2005.
[9] B. L. Kumova, "Dynamically adaptive partition-based data distribution management," presented at the the 2005 Workshop on Principles of Advanced and Distributed Simulation, 2005.
[10] O. Eroglu, H. A. Mantar, and F. E. Sevilgen, "Quadtree-based approach to data distribution management for distributed simulations," presented at the the 2008 Spring simulation multiconference, Ottawa, Canada, 2008.
[11] G. Tan, X. Liang, F. Moradi, and Z. Yusong, "An agent-based DDM filtering mechanism," presented at the the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000.
[12] G. Tan, X. Liang, F. Moradi, and S. Taylor, "An agent-based DDM for High Level Architecture," presented at the the 15th Workship on Parallel and Distributed Simulation, 2001.
[13] W. Lihua, S. J. Turner, and W. Fang, "Interest management in agent-based distributed simulations," presented at the the Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2003.
[14] Y. Jun, C. Raczy, and G. Tan, "Evaluation of a sort-based matching algorithm for DDM," presented at the the sixteenth workshop on Parallel and distributed simulation, Washington, D.C., 2002.
[15] K. Pan, S. J. Turner, W. Cai, and Z. Li, "An Efficient Sort-Based DDM Matching Algorithm for HLA Applications with a Large Spatial Environment," presented at the the 21st International Workshop on Principles of Advanced and Distributed Simulation, 2007.
[16] C. Raczy, G. Tan, and J. Yu, "A sort-based DDM matching algorithm for HLA," *ACM Transactions on Modeling and Computer Simulation*, vol. Volume 15 Issue 1, pp. 14-38, 2005.
[17] P. Gupta and R. K. Guha, "A Comparative Study of Data Distribution Management Algorithms," *The Journal of Defense Modeling and Simulation on Applications, Methodology, Technology*, vol. Volume 4 Issue 2, pp. 127-146, 2007.
[18] C. Raczy, G. Tan, and J. Yu, "A sort-based DDM matching algorithm for HLA," *ACM Trans. Model. Comput. Simul.*, vol. 15, pp. 14-38, 2005.
[19] J. Y. C Raczy, G Tan, SC Tay, R Ayani, "Adaptive data distribution management for HLA RTI," presented at the the 2002 European Simulation Interoperability, 2002.
[20] A. Boukerche and A. J. Roy, "Dynamic Grid Based Multicast Group Assignment in Data Distribution Management," in *Proceeding Fourth IEEE International Workshop on Distributed Simulation and Real-time Applications Workshop 2000*, 2000, pp. 27-34.
[21] A. Boukerche and A. J. Roy, "Dynamic Grid-Based Approach to Data Distribution Management," *Journal of Parallel and Distributed Computing*, pp. 366-392, 2002.
[22] A. Boukerche, C. Dzermajko, and K. Lu, "Dynamic Grid-Based vs Region-Based Data Distribution Management in Multi-Resolution Large-Scale Distributed Systems," in *Proceedings of the 18th International Parallel & Distributed Processing Symposium*, 2004.
[23] S. J. Rak and D. J. V. Hook, "Evaluation of Grid-Based Relevance Filtering for Multicast Group Assignment," presented at the the 1996 Distributed Interactive Simulation, 1996.
[24] A. Boukerche, G. YunFeng, and R. B. Araujo, "An adaptive dynamic grid-based approach to data distribution management," presented at the the 20th International Parallel and Distributed Processing Symposium, 2006.
[25] A. Boukerche, Y. Gu, and R. B. Araujo, "Performance Analysis of an Adaptive Dynamic Grid-Based Approach to Data Distribution Management," presented at the the tenth IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2006.
[26] A. Boukerche and Y. Gu, "An Efficient Adaptive Transmission Control Scheme for Large-Scale Distributed Simulation Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. Volume 20 Issue 2, pp. 246-260, 2009.
[27] S. H. Lo, C. A. Chiu, D. Y. Hong, F. P. Pai, and Y. C. Chung, "MGRID: A Modifiable-Grid Region Matching Approach for DDM in the HLA RTI," presented at the the 2009 Sping Simulation Multiconference, San Diego, CA, 2009.
[28] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-Level Multicast Using Content-Addressable Networks," in *Networked Group Communication*. vol. LNCS 2233, ed: Springer Berlin / Heidelberg, 2001, pp. 14-29.

Biography

Shih-Hsiang Lo received his BS from the Department of Computer Science, National Chengchi University, Taipei, Taiwan, in 2004, and MS from the Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently a PhD student in the Department of Computer Science at National Tsing Hua University. His research interests are in the areas of modeling, simulation, parallel and distributed computing, and game server design.

Yeh-Ching Chung received a BS in Information Engineering from Chung Yuan Christian University in 1983, and the MS and PhD in Computer and Information Science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an Associate Professor in 1992 and became a Full Professor in 1999. From 1998 to 2001, he was the Chairman of the Department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a Full Professor. His research interests include parallel and distributed processing, cluster systems, grid computing, multi-core tool chain design, and multi-core embedded systems. He is a Member of the IEEE computer society and ACM.