Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

IMPLEMENTATION OF DATA DISTRIBUTION MANAGEMENT SERVICES IN A SERVICE ORIENTED HLA RTI

Ke Pan
Stephen John Turner
Wentong Cai
Zengxiang Li

Parallel and Distributed Computing Center
School of Computer Engineering
Nanyang Technological University
Singapore 639798

ABSTRACT

Simulation is a low cost and safe alternative to solve complex problems in various areas. To promote reuse and interoperability of simulation applications and link geographically dispersed simulation components, distributed simulation was introduced. The High Level Architecture (HLA) is the IEEE standard for distributed simulation. The actual implementation of the HLA standard is provided by a Run Time Infrastructure (RTI). The HLA defines six service groups. Its Data Distribution Management (DDM) service group aims at optimizing communication efficiency between simulation components and there are various approaches for DDM implementation. We have previously developed a Service Oriented HLA RTI (SOHR) which enables distributed simulations to be conducted across administrated domains on the Grid. It maps the six HLA service groups into different management services and creates a plug-and-play paradigm for an HLA RTI implementation so that different approaches for a service group can be easily implemented into SOHR. To demonstrate the plug-and-play paradigm, this paper discusses the implementation of two DDM approaches, the grid-based approach and an extended efficient sort-based approach, in SOHR. Experiments have also been carried out to compare their performance in different scenarios.

1 INTRODUCTION

Simulation is a low cost and safe alternative to solve complex problems in various areas such as production, business, education, science and engineering. To promote reuse and interoperability of simulation applications and link geographically dispersed simulation components, distributed simulation (Fujimoto 2000) was introduced. The High Level Architecture (HLA), originally proposed by the Defense Modeling and Simulation Office (DMSO), was revised as IEEE 1516 standard for distributed simulation in September 2000 (IEEE 2000). While the HLA defines the rules, interface specification and OMT (Object Model Template), a Run Time Infrastructure (RTI), such as the DMSO RTI (DMSO 2002), provides the actual implementation of the HLA standard. In HLA terminology, a distributed simulation application is called a federation which comprises of several simulation components called federates. The RTI is a communication middleware which provides several groups of services used by federates to communicate with each other in the same federation. There are totally six service groups provided, namely Federation Management, Declaration Management, Object Management, Ownership Management, Time Management and Data Distribution Management (DDM). DDM aims at optimizing communication efficiency between federates.

DDM provides a set of services to reduce message traffic over the network and irrelevant processing by receiving federates (Morse and Steinman 1997). It allows data producers (sending federates) to specify their update regions in an universal multi-dimensional routing space (IEEE 2000, Morse and Petty 2001). It also allows data consumers (receiving federates) to specify their subscription regions in the same universal multi-dimensional routing space. Data are directed from sending federates to receiving federates if and only if there is an overlap between the update and subscription regions. A region is specified as a range on each dimension of a subset of all dimensions in the universal multi-dimensional routing space. An update region and a subscription region overlap if and only if they have common dimensions and their ranges overlap on each common dimension. An example is shown in Figure 1. There is one update region U1 and two subscription regions S1 and S2. All the three regions are specified using dimensions X and Y in the universal routing space. U1 overlaps with S1 because their ranges overlap on both dimensions X and Y. U1 does not overlap with S2 because there is no overlap

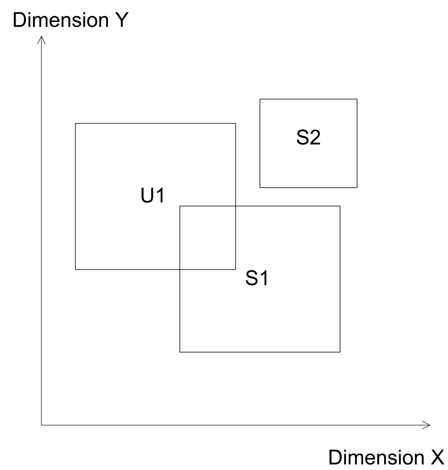


Figure 1: An example of overlapping regions

between their ranges on dimension X. This means that the federate associated with S1 will receive data from U1 but the federate associated with S2 will not.

We have previously developed a Service Oriented HLA RTI (SOHR) which enables distributed simulations to be conducted across administrated domains on the Grid (Pan et al. 2007b). It maps the six HLA service groups into different management services and creates a plug-and-play paradigm for an HLA RTI implementation so that different approaches for a service group can be easily implemented into SOHR. To demonstrate the plug-and-play paradigm, this paper discusses the implementation of two DDM approaches, the grid-based approach and an extended efficient sort-based approach, in SOHR. Experiments have also been carried out to compare their performance in different scenarios.

The rest of the paper is organized as follows. Section 2 describes major DDM approaches and their differences. Section 3 briefly introduces SOHR. Then Section 4 describes the two DDM implementations in SOHR and their performance is compared experimentally in Section 5. Finally, Section 6 concludes the whole paper.

2 MAJOR DDM APPROACHES

There are various DDM approaches and their differences can be clarified by decomposing the DDM computation into the following four distinct processes (Petty and Morse 2004).

- Declaring: The federates create, modify or delete their respective update and subscription regions throughout a federation execution as their data requirements change. A change is originated from a federate and other parts of the federation need to be informed of it for the next process, matching.
- Matching: The RTI derives the overlaps between update and subscription regions with a matching algorithm.
- Connecting: Based on the result of the matching process, the RTI establishes network data flow connectivity depending on the network infrastructure.
- Routing: The RTI transports data from the sending federates to the appropriate receiving federates using the connectivity established during the connecting process.

For the discussion in this paper, we assume a sender (producer) side data filtering mechanism. Subscription region information is sent from data consumers to data producers in the declaring process. With the received subscription region information and their own update region information, data producers do the matching. The matching result is used to guide the later connecting and routing processes.

2.1 Region-Based Approach

The region-based approach is also known as the brute force approach. Each subscription region modification results in the sending of a region update message in its declaration process. Its matching process checks each update region with each subscription region to derive the overlapping information and therefore has $O(N^2)$ computational complexity, where N denotes the total number of regions, which is not scalable with respect to N. Since its matching derives exact overlapping information, its routing process does not involve irrelevant data and therefore is efficient. The region-based approach is used in an early version of DMSO RTI 1.3 (Van Hook and Calvin 1998), and the MAK high performance RTI (Wood 2002).

2.2 Grid-Based Approach

The grid-based approach divides a routing space into a grid of cells. Each region is then mapped to the grid cells. An update region and a subscription region are assumed to overlap with each other if and only if they share at least one common grid cell. DMSO RTI 1.3 NG provides an implementation of this approach in its StaticGridPartitioned strategy (Hyett and Wuerfel 2002).

Upon a subscription region modification, the declaration process checks whether the set of cells with which the region overlaps changes or not. Only when there is a change, communication cost is incurred for sending region updates. Its matching process is simple but not exact. Thus, its routing process involves irrelevant data sending. A specific implementation, such as when a multicast group is used per grid cell, may also bring the issue of duplicated data sending when two regions share more than one grid cell. Both irrelevant and duplicated data sending incur not only communication cost but also receiver-side filtering cost. A larger grid cell size causes more irrelevant data to be exchanged and therefore more communication cost (Tan et al. 2000, Rak and Van Hook 1996). On the other hand, a smaller grid cell size results in more grid cells, which increases both the declaration cost and the matching cost (Tan et al. 2000). Ayani et al. have carried out a study on optimization of the cell size for the grid-based approach (Ayani et al. 2000).

2.3 Hybrid Approach

Tan proposed a hybrid approach which utilizes ideas of both the region-based and grid-based approaches (Tan et al. 2000). Each subscription region modification results in the sending of a region update message in its declaration process. Its matching process first uses the grid-based approach to map all regions to the grid cells. Then the region-based approach is used to do exact matching between update regions and subscription regions for each grid cell. In this way, the matching cost is reduced compared with the pure region-based approach and its routing cost is reduced compared with the pure grid-based approach. However, its matching cost depends on the chosen size of grid cells. The grid-filtered region-based approach proposed by Boukerche et al. (Boukerche et al. 2005) is a variation of the hybrid approach.

2.4 Sort-Based Approach

Raczy proposed a sort-based approach (Raczy et al. 2005). For each dimension, it first inserts bounds of all regions into a list and sorts it. Then it checks the overlapping status by scanning all the sorted lists once. The computational complexity of matching is still $O(N^2)$, but its utilization of bit operations makes it very promising. However, it has three major drawbacks (Pan et al. 2007a). It requires large storage space which is not scalable with respect to the number of regions. It uses costly row update operations of a matrix and lacks a dynamic mechanism for selective region modifications.

To overcome the drawbacks of Raczy's sort-based algorithm, our group proposed an efficient sort-based approach for HLA applications with a large spatial environment (Pan et al. 2007a). Instead of sorting all region bounds on a dimension into a list, lower bounds and higher bounds of update regions and subscription regions are separately sorted into four lists. For the matching, instead of scanning all lists, the bounds of a region are mapped on relevant lists for deriving the overlapping status. This matching algorithm has been proven to be more efficient than Raczy's algorithm in a large spatial environment. It requires less storage space and is able to dynamically deal with selective region modifications.

Similar to the region-based approach, for both the two sort-based algorithms, each subscription region modification results in the sending of a region update message in the declaration process. The matching is exact so the routing process is efficient.

3 SOHR FRAMEWORK

We have previously developed a Service Oriented HLA RTI (SOHR) (Pan et al. 2007b). It implements an HLA RTI using Grid services so that distributed simulations can be conducted across administrative domains. Figure 2 shows the architecture of the SOHR framework. It contains seven key Grid services, namely the RTI Index Service, the LS (Local Service) and five management services, all of which are implemented based on GT4. All services except the RTI Index Service follow the WS-Resource factory design pattern (Sotomayor 2005). In the WS-Resource factory design pattern, information is organized into resource instances; a resource home keeps track of the multiple resource instances; a factory service is defined to create

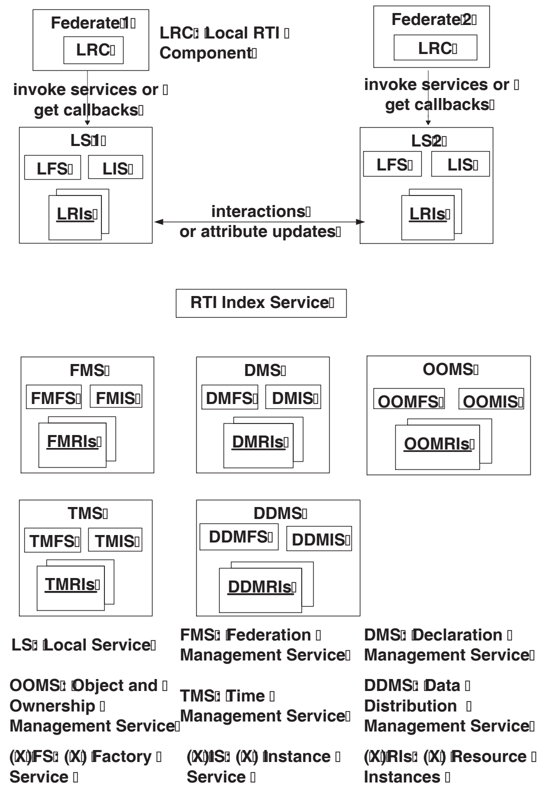


Figure 2: Framework overview

new resource instances; an instance service uses the resource home to find the client specified resource instance and operates on it.

The RTI Index Service provides a system-level registry so that all other services are able to register their EPRs (End Point References) here and dynamically discover each other. It also provides services to create and destroy federations in the system.

The five management services correspond to the six HLA service groups (Note: For the convenience of maintaining object instances and their respective attributes, the Object Management and Ownership Management service groups are combined into a single OOMS.). Each of them consists of a factory service, an instance service and multiple resource instances, with one resource instance for each federation. For example, the DDMS provides functionalities of the HLA Data Distribution Management service group as a Grid service and consists of the factory service DDMFS, the instance service DDMIS and multiple resource instances DDMRIs, with one DDMRI for each federation.

The LS (Local Service) is used as a messaging broker of federates and consists of the factory service LFS, the instance service LIS and multiple resource instances LRIs, with one LRI for each federate. Multiple federates may share the same LS, but each has its own LRI. A federate communicates with the outside world through its LRI by invoking services and getting callbacks. As shown in Figure 3, the LRI is structured into six modules. Each of the modules except the Callback Module corresponds to one of the management services. The Callback Module is used to buffer callbacks for a federate.

The objective of separating the HLA service groups into different Grid services and using a modular structure for the LRI is to create a plug-and-play paradigm for an HLA RTI implementation so as to build an extensible SOHR framework. There may be multiple algorithms for the implementation of an HLA service group, and these can generally be classified as centralized algorithms and distributed algorithms. In a centralized algorithm, the major processing is done by the management service while the corresponding module in the LRI simply keeps some necessary information related to its federate. In a distributed algorithm, the major processing is done by the module in the LRI while the corresponding management service simply keeps relevant centralized information of the federation. A particular module in the LRI and its corresponding management service cooperate to provide the services of the HLA service group. Based on different algorithms for the HLA service group, multiple combinations of a particular module of the LRI and its corresponding management service

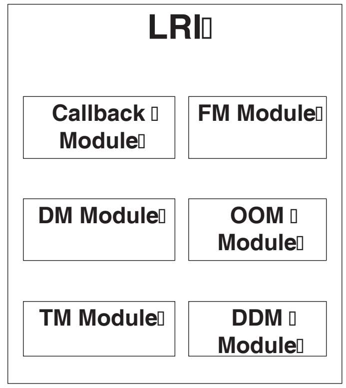


Figure 3: LRI structure

can be implemented and plugged into SOHR. Users are able to choose a specific algorithm of an HLA service group for a federation through a federate-side configuration file. After the LRI is created by the factory service LFS, the particular module of the LRI for the chosen algorithm will be generated and its corresponding management service will be used. To demonstrate the plug-and-play paradigm, Section 4 describes the implementation of two DDM approaches in SOHR.

The LRC (Local RTI Component) is a federate's local library that implements the HLA service interfaces (IEEE 2000) and simply does the translation between HLA service interfaces and the corresponding LIS Grid service invocations.

4 DDM IMPLEMENTATION IN SOHR

We have implemented the grid-based approach and an extended efficient sort-based approach in SOHR. Both the two approaches have been implemented in a distributed manner, so that the major DDM work is done by LRI's DDM module while the DDMS is simply responsible for generating region handles and keeping an association between federates and their respective regions. The association information is fetched by a newly joining federate for its LRI's DDM module initialization. As the matching process only derives the overlapping information between regions, the association information is necessary for a sending federate to derive the list of receiving federates for its data transmission (connecting). The association information is updated whenever a new region is created or an old regions is deleted during the whole federation execution. For simplicity, data messages are sent in a peer-to-peer way for the routing process in our implementations.

4.1 Grid-Based DDM Implementation

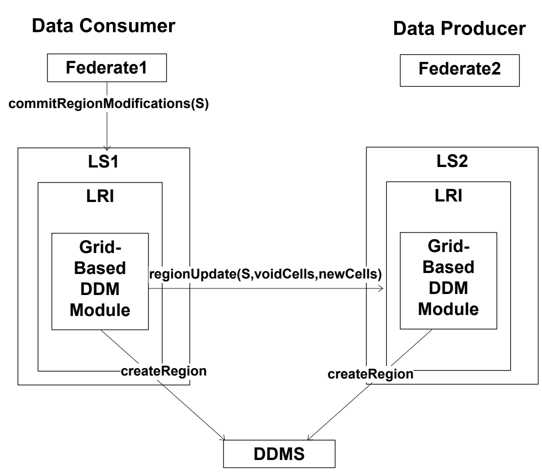


Figure 4: Grid-based DDM implementation

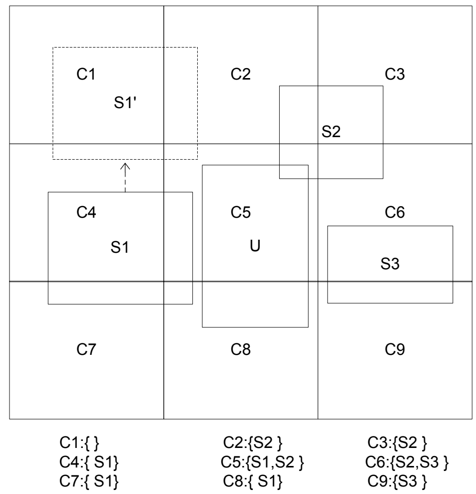


Figure 5: An example for the grid based DDM implementation

The grid-based DDM approach was implemented by incorporating a Grid-Based DDM Module in the LRI structure as shown in Figure 4. For each grid cell, the Grid-Based DDM Module maintains a list of remote subscription regions which overlap with the cell. When an attribute update or interaction is requested to be sent out with a local update region, the data message is sent to all federates associated with regions in the remote subscription lists of the cells with which the local update region overlaps (routing). An example is shown in Figure 5. There are nine grid cells with their respective remote subscription lists. U is a local update region while S1, S2 and S3 are remote subscription regions. When an attribute update is requested to be sent out with U, the federates associated with regions in the union of C5 and C8's remote subscription region list, i.e. S1 and S2, will receive the attribute update. As the grid-based matching is not exact, there will be irrelevant data message exchange, e.g. U and S1 in Figure 5. To further enable a receiver-side filtering mechanism, the sending update region information is piggybacked in the data message. It is checked against the subscription region information at the receiver side before the data message can be delivered to the receiving federate.

As shown in Figure 4, when a subscription region modification is committed (declaring) at the data consumer side, if there is any overlapping status change, other federates' corresponding Grid-Based DDM Modules need to be informed through a region update message. The message should specify the modified subscription region handle, the void cells which overlapped with the region before the modification but do not overlap with the region after the modification, the new cells which did not overlap with the region before the modification but overlap with the region after the modification. Upon receiving a region update message, the Grid-Based DDM Module at the data producer side removes the specified subscription region from the remote subscription region lists of the void cells and adds the specified subscription region into the remote subscription region lists of the new cells. Each of these list updates is referred to as a List Update (LU) operation. For example, in Figure 5, if S1 changes its position to $S1'$, the region update message is in the form of regionUpdate(S1,{C7,C8},{C1,C2}). This triggers four LU operations.

4.2 Extended Efficient Sort-Based DDM Implementation

Upon a subscription region modification, our original efficient sort-based algorithm abandons all old overlapping information related to the modified region, rematches the modified region against all update regions and thus derives the new overlappping information related to the modified region (Pan et al. 2007a). However, abandoning all old overlapping information is not necessary as some are still valid after the region modification, especially when the moving distance of the region is relatively small compared with the region size. To improve the matching performance further, the efficient sort-based algorithm has been extended to incrementally match for dynamic region modifications. Since the matching algorithm is not the focus of this paper, its details are not described here.

The extended efficient sort-based approach was implemented by incorporating an Extended Efficient Sort-Based DDM Module in the LRI structure as shown in Figure 6. A sender-side filtering mechanism is adopted, so that the boundary information of subscription regions is sent between the federates' LRIs (declaring) and the Extended Efficient Sort-Based

DDM Module is responsible for matching based on the received boundary information. Since the extended efficient sort-based matching is exact, further receiver-side filtering is not needed for data message sending (routing).

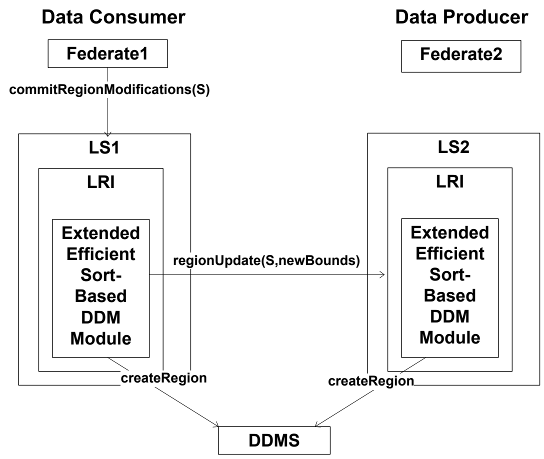


Figure 6: Extended efficient sort-based DDM implementation

5 EXPERIMENTS AND RESULTS

To compare the performance of the two DDM implementations, experiments were carried out with a time-stepped federation of moving tanks. It contains two federates each of which simulates 50 moving tanks in a 40km*40km two-dimensional routing space. In each time step of 1 minute simulation time, a tank has a certain probability of moving in a random direction (north, south, east or west). If a tank moves in a time step, it moves with a constant speed. A tank has an update region and a subscription region surrounding its current position, and it sends its status update though an attribute update message in each time step. The whole federation lasts for a simulation time of 100 minutes. All the following experiments were executed in a cluster with a 10Gb/s infiniBand connection. Each node of the cluster is installed with two dual core Xeon 3.0GHz CPUs, a 4GB RAM and a Redhat Enterprise Linux 4 OS. One federate with its LS were executed on one node and the other federate with its LS were executed on another node. All the other SOHR services were executed on a third node.

5.1 Performance with Basic Setup

The basic experimental setup has the tank moving probability equal to 0.25, the tank moving speed equal to 1000 meters per minute, the update region size equal to 200m*200m and the subscription region size equal to 2000m*2000m. The performance results are shown in Table 1.

For the extended efficient sort-based approach, the number of Region Update Messages (RUM) is close to 2500, which is as expected since the moving probability is 0.25 and there are 100 tanks and 100 time steps of 1 minute. Since its matching is exact, the Attribute Update Message Valid Percentage (AUMVP) is 100%.

As the performance of the grid-based approach depends on the chosen size of the grid cells, we have varied the number of grid cells from 1*1 to 60*60. The results show that the number of RUM increases with an increasing number of cells, and becomes saturated when the number of cells approaches 30*30. This is because, as the size of a grid cell decreases towards the distance moved by a tank in one time step, the possibility that a tank movement causes a change of its subscription region's overlapping grid cells becomes larger and finally approaches 1. Other points to note with an increasing number of cells include the increasing number of List Update (LU) operations, the decreasing number of exchanged Attribute Update Messages (AUM) and the increasing AUMVP. As the number of cells increases to 20*20, the total number of messages, which is the summation of RUM and AUM, decreases due to the decreasing AUM. This causes the total simulation time to decrease. As the number of cells increases further, the total number of messages may become lower (e.g. the 50*50 case) due to the increasing AUMVP and is expected to approach the total number of messages of the extended efficient sort-based approach. However, the simulation time increases due to the increasing number of list update operations.

Table 1: Simulation results of the basic setup

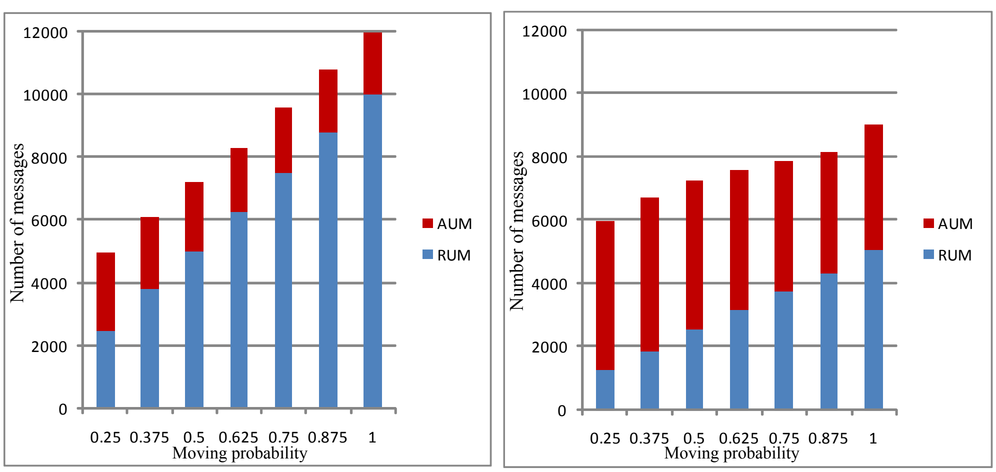
DDM Approach	RUM	LU	AUM	AUMVP	Message Total	Time (S)
Extended Efficient	2483	-	2504	100%	4987	122
Grid-Based 1*1	0	0	10000	26.04%	10000	145
Grid-Based 2*2	225	421	10000	29.17%	10225	146
Grid-Based 4*4	466	1895	9970	27.97%	10436	149
Grid-Based 8*8	927	7331	7913	34.02%	8840	143
Grid-Based 10*10	1195	11394	7193	38.57%	8388	139
Grid-Based 20*20	1283	43986	4695	52.03%	5978	129
Grid-Based 30*30	2436	97674	3876	62.46%	6312	135
Grid-Based 40*40	2446	172546	3926	70.12%	6372	145
Grid-Based 50*50	2431	267940	3436	76.51%	5867	150
Grid-Based 60*60	2449	385283	3574	77.95%	6023	164

RUM: Region Update Message LU: List Update AUM: Attribute Update Message
AUMVP: Attribute Update Message Valid Percentage

In the basic experiment setup, the extended efficient sort-based approach is always better than the grid-based approach and the grid-based approach has best performance when the number of cells is set to 20*20. In the following experiments, the number of cells is chosen to be 20*20 for the grid-based approach.

5.2 Performance with Respect to the Moving Probability

To analyze how the tank moving probability affects the performance of the two DDM implementations, we have increased it from 0.25 to 1 in increment steps of 0.125 based on the basic experiment setup in Subsection 5.1. The performance in terms of the number of messages is shown in Figure 7 and the performance in terms of the simulation time is shown in Figure 8. For both approaches, the total number of messages and simulation time increase with an increasing moving probability mainly due to the increasing number of RUM exchanged. For the grid-based approach, an increase in the number of RUM messages increases the number of LU operations performed, which also has some effect on increasing the simulation time. However, the extended efficient sort-based approach is affected more by the increasing moving probability as it exchanges more RUM, so the grid-based approach begins to outperform the extended efficient sort-based approach when the moving probability is increased to 0.625 in both figures.



(a) Extended Efficient Sort-Based (b) Grid-Based

Figure 7: Number of messages with respect to the moving probability

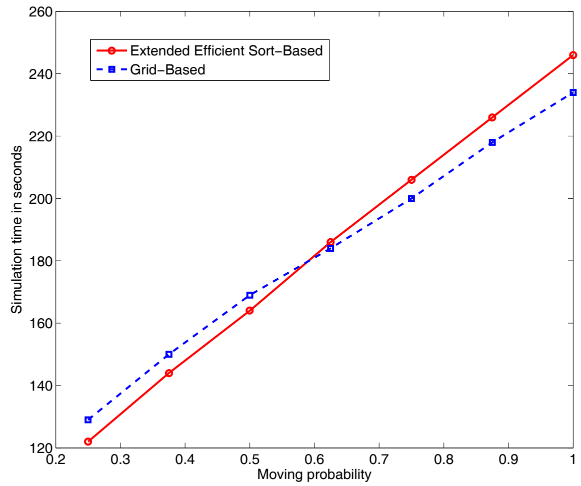
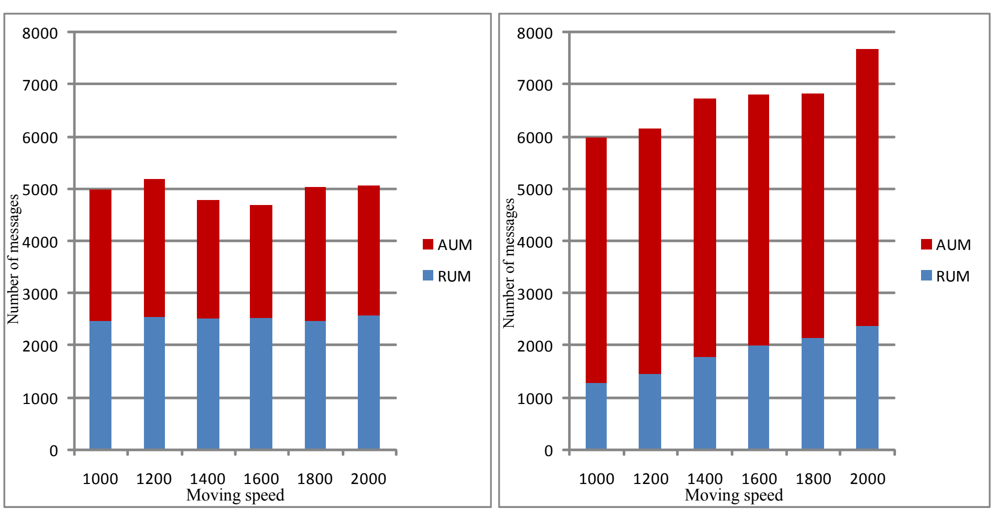


Figure 8: Simulation time with respect to the moving probability

5.3 Performance with Respect to the Moving Speed

To analyze the performance of the two DDM implementations with respect to the tank moving speed, we have varied the moving speed from 1000 meters per minute to 2000 meters per minute based on the basic experiment setup in Subsection 5.1. Figure 9 shows the performance in terms of the number of messages and Figure 10 shows the performance in terms of the simulation time. As we can see, the number of RUM messages of the extended efficient sort-based approach does not vary much with an increasing moving speed as a RUM message always needs to be sent for each subscription region modification no matter how much the region moves. As the moving speed does not have much effect on the number of AUM messages, the performance of the extended efficient sort-based approach does not vary much. In contrast, the total number of messages of the grid-based approach increases, as an increasing moving speed increases the possibility that a tank movement changes its subscription region's overlapping grid cells and thus increases the number of RUM exchanged. For this reason and the fact that an increasing number of RUM messages also increases the number of LU operations, the simulation time of the grid-based approach increases with an increasing moving speed.



(a) Extended Efficient Sort-Based

(b) Grid-Based

Figure 9: Number of messages with respect to the moving speed

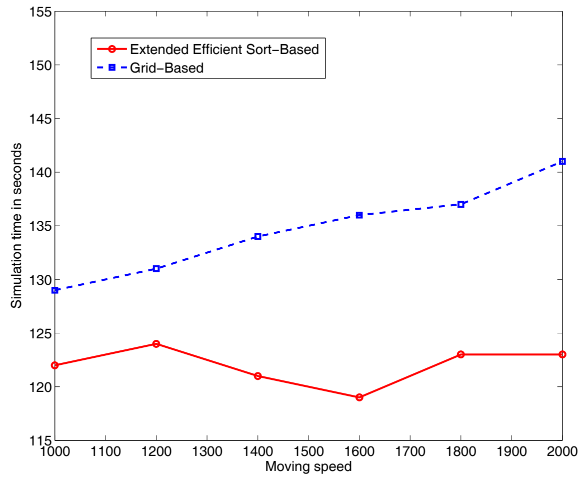
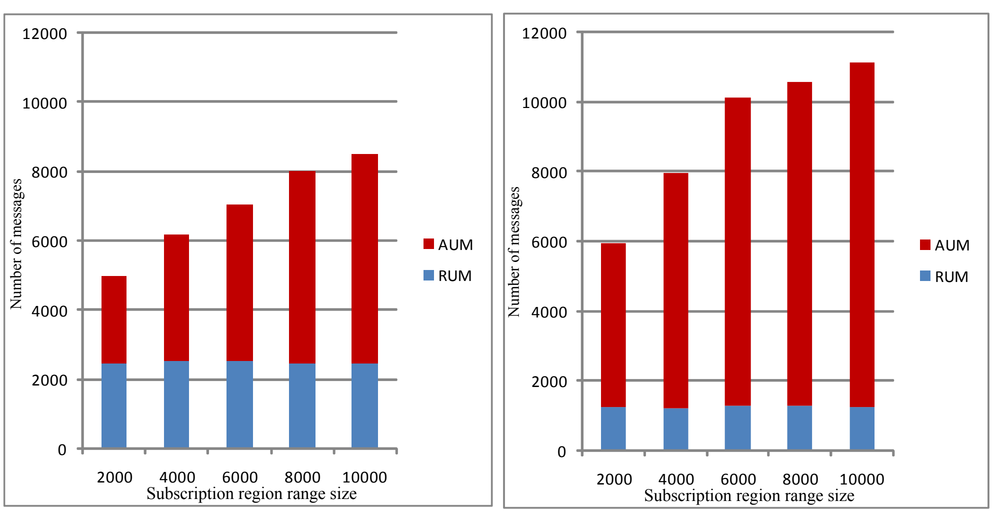


Figure 10: Simulation time with respect to the moving speed

5.4 Performance with Respect to the Subscription Region Size

Based on the basic experiment setup in Subsection 5.1, we have also increased the subscription region range size from 2000 meters to 10000 meters in increment steps of 2000 meters. The performance in terms of the number of messages is shown in Figure 11 and the performance in terms of simulation time is shown in Figure 12. As we can see, for both approaches, the total number of messages and simulation time increase with an increasing subscription region range size mainly due to the increasing number of AUM sent. As the grid-based approach sends more AUM due to its inexact matching, its total number of messages is affected more by the subscription region size. The increasing subscription region range size also increases the LU operations performed by the grid-based approach. For these two reasons, the simulation time of the grid-based approach is affected more by the increasing subscription region range size.

Of the three parameters we have varied, the moving probability affects the simulation time most. This is because, a tank movement involves several RTI service invocations, including the requests of setting the range bounds for each dimension and committing region modifications, and each request is a separate Grid service invocation in SOHR.



(a) Extended Efficient Sort-Based (b) Grid-Based

Figure 11: Number of messages with respect to subscription region range size

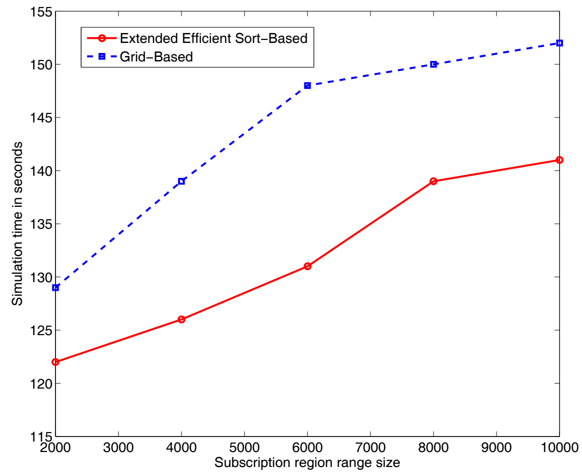


Figure 12: Simulation time with respect to subscription region range size

6 CONCLUSION

The HLA is the IEEE standard for distributed simulation and it defines six service groups. The DDM service group aims at optimizing communication efficiency between simulation components and there are various approaches for DDM implementation. We have previously developed a Service Oriented HLA RTI (SOHR) which enables distributed simulations to be conducted across administrated domains on the Grid. It maps the six HLA service groups into different management services and creates a plug-and-play paradigm for an HLA RTI implementation so that different approaches for a service group can be easily implemented into SOHR. To demonstrate the plug-and-play paradigm, this paper discusses the distributed implementation of two DDM approaches, the grid-based approach and an extended efficient sort-based approach, in SOHR. Experiments have also been carried out to compare their performance in different scenarios. Results show that the extended efficient sort-based approach is more effective at reducing the number of AUM sent, but at the expense of extra RUM. If regions are modified frequently as in Subsection 5.2, the grid-based approach can give better performance with a well chosen number of cells.

REFERENCES

Ayani, R., F. Moradi, and G. Tan. 2000. Optimizing cell-size in grid-based DDM. In *Proceedings of the 14th Workshop on Parallel and Distributed Simulation*, 93–100.

Boukerche, A., N. J. Mcgraw, and R. B. Araujo. 2005. A grid-filtered region-based approach to support synchronization in large-scale distributed interactive virtual environments. In *Proceedings of the 2005 International Conference on Parallel Processing Workshops*, 525–530.

DMSO 2002. *High level architecture rti 1.3ng programmer's guide version 5*.

Fujimoto, R. M. 2000. *Parallel and distributed simulation systems*. Wiley Interscience.

Hyett, M., and R. Wuerfel. 2002. Implementation of the data distribution management services in the RTI-NG. In *Proceedings of the 2002 Spring Simulation Interoperability Workshop, paper no. 02S-SIW-044*.

IEEE 2000. *Standard 1516 (hla rules), 1516.1 (federate interface specification) and 1516.2 (object model template)*.

Morse, K. L., and M. D. Petty. 2001. Data distribution management migration from DoD 1.3 to IEEE 1516. In *Proceedings of the 5th IEEE International Workshop on Distributed Simulation and Real Time Applications*, 58–65.

Morse, K. L., and J. S. Steinman. 1997. Data distribution management in the HLA: Multidimensional regions and physically correct filtering. In *Proceedings of the 1997 Spring Simulation Interoperability Workshop, paper no. 97S-SIW-052*.

Pan, K., S. J. Turner, W. Cai, and Z. Li. 2007a. An efficient sort-based DDM matching algorithm for HLA applications with a large spatial environment. In *Proceedings of the 21st ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*, 70–82.

Pan, K., S. J. Turner, W. Cai, and Z. Li. 2007b. A service oriented HLA RTI on the grid. In *Proceedings of the IEEE International Conference on Web Services*, 984–992.

Petty, M. D., and K. L. Morse. 2004. The computational complexity of the high level architecture data distribution management matching and connecting processes. *Simulation Modeling Practice and Theory* 12 (3-4): 217–237.

Raczy, C., G. Tan, and J. Yu. 2005. A sort-based DMM matching algorithm for HLA. *ACM Transactions on Modeling and Computer Simulation* 15 (1): 14–38.

Rak, S. J., and D. J. Van Hook. 1996. Evaluation of grid-based relevance filtering for multicast group assignment. In *Proceedings of the Distributed Interactive Simulation*, 739–747.

Sotomayor, B. 2005. *The globus toolkit 4 programmer's tutorial*. http://gdp.globus.org/gt4-tutorial/.

Tan, G., R. Ayani, Y. Zhang, and F. Moradi. 2000. Grid-based data management in distributed simulation. In *Proceedings of the 33rd Annual Simulation Symposium*, 7–13.

Tan, G., Y. Zhang, and R. Ayani. 2000. A hybrid approach to data distribution management. In *Proceedings of the 4th IEEE International Workshop on Distributed Simulation and Real-Time Applications*, 55–61.

Van Hook, D. J., and J. O. Calvin. 1998. Data distribution management in RTI 1.3. In *Proceedings of the 1998 Spring Simulation Interoperability Workshop, paper no. 98S-SIW-206*.

Wood, D. D. 2002. Implementation of DDM in the MAK high performance RTI. In *Proceedings of the 2002 Spring Simulation Interoperability Workshop, paper no. 02S-SIW-056*.

AUTHOR BIOGRAPHIES

KE PAN is a Ph.D. student in the School Of Computer Engineering (SCE) at the Nanyang Technological University (NTU) of Singapore. He received his B.Eng. in the same school with a 1st class honor degree. His email address is <pank0001@ntu.edu.sg>.

STEPHEN JOHN TURNER is a Professor of Computer Science in SCE at NTU (Singapore) and Head of the Computer Science Division. He is steering committee chair of the PADS conference and advisory committee member of the DS-RT symposium. He is also an area editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS). His email address is <assjturner@ntu.edu.sg>.

WENTONG CAI is a Professor of Computer Science in SCE at NTU (Singapore) and Director of the Parallel and Distributed Computing Center. He is currently an associate editor of ACM Transactions on Modeling and Computer Simulation (TOMACS), editorial board member of multiagents and Grid Systems - An International Journal, and editorial board member of International Journal of Computers and Applications. His email address is <aswtcai@ntu.edu.sg>.

ZENGXIANG LI is a Ph.D. student in SCE at NTU (Singapore). He received his M.Eng. from the Shanghai Jiao Tong University (China). His email address is lize0001@ntu.edu.sg>.