Layer Partition-based Matching Algorithm of DDM

Nwe Nwe Myint Thein, and Nay Min Tun

***Abstract*—**High Level Architecture (HLA) is architecture for reuse and interoperation of simulations. In HLA paradigm, the Runtime Infrastructure (RTI) provides a set of services. Data Distribution Management (DDM) service reduces message traffic over the network. DDM aims to control and limit the data exchanged between federates during federation. Each federate may inform the RTI about its intention to publish some data or it may subscribe to receive a subset of the published data. DDM services are used to reduce the transmission and receiving of irrelevant data and aimed at reducing the communication over the network. These services rely on the computation of the intersection between "update" and "subscription" regions. When calculating the intersection between update regions and subscription regions, the higher computation overhead can occur. Currently, there are several main DDM filtering algorithms. This paper proposes the layer partition-based matching algorithm for DDM in the HLA-based large-scale distributed simulations. The new algorithm chooses the dynamic pivot based on regions distribution in the routing space. The binary partition-based algorithm is fundamentally based on a divide and conquers approach. This algorithm always chooses the midpoint as the pivot point of routing space. This approach promises low computational overhead, since it does not require unnecessary comparisons within regions in different partitions. The proposed algorithm firstly calculates the regions distribution. Then, the partitioning among regions performs based on the result of choosing pivot based on region detection and defines the matching area that entirely covers all regions which need to match with regions at pivot point. The proposed algorithm provides the more definite matching area between update region and subscription region during matching process. This algorithm guarantees low computational overheads for matching process based on the overlapping degree between the regions and reduce the irrelevant message among federates.

***Keywords*—**HLA, DDM, Region-based Algorithm, Grid-based Algorithm, Sort-based Algorithm, Binary Partition-based Algorithm, Layer Partition-based Algorithm**.**

I. Introduction

THE High Level Architecture (HLA) is architecture for reuse and interoperation of simulations. HLA provides the specification of a common technical architecture for use across all classes of simulations.

Nwe Nwe Myint Thein, University of Technology (Yatanarpon Cyber City), PyinOoLwin, email id: nwenwemyintthein@gmail.com,

Nay Min Tun, Computer University (Kyaing Tong), email id: naymin300777@gmail.com

HLA's Run Time Infrastructure (RTI) is a software component that provides commonly required services to simulation systems. There are several groups of services, which are provided by the HLA Runtime Infrastructure (RTI) to coordinate the operations and the exchanges of data between federates (simulations) during a runtime execution. The interaction of object instances across user applications is supported by the function of RTI, which is similar to a distributed operating system. The High Level Architecture (HLA) specifications define six services including DDM services supported by the Runtime Infrastructure (RTI). HLA DDM service's goal is to limit the messages received by federates in large distributed federations to those messages of interest in order to reduce the data set required to be processed by the receiving federate and the message traffic over the network. DDM is one of the most filtering mechanisms regarding large scale distributed simulations. [5] These services rely on the computation of the intersection between "update" and "subscription" regions. So its capability is limited by the computational overhead for calculating the intersection between update regions and subscription regions which represent the data producers and the data consumers. [11] Several DDM matching schemes have proposed in the literature. These matching algorithms do not take into account how much regions are generated and distributed in the multidimensional space. The region distribution is the important factor for the matching calculation between data producers and data consumers.

In this paper, we propose a new layer partition-based algorithm and calculating the particular matching area for the specified partitioning point. These algorithm and method based on the binary partition-based matching algorithm in order to reduce the computational overhead of the matching process. They also intend to ease the irrelevant message routing over the network. This algorithm firstly detects the position of regions and calculates the region distribution status according to their coordinates. The proposed algorithm partitioned the routing space based on the dynamic pivot point. This point is defined one point where the most updaters and subscribers are converged. Then the proposed algorithm need to specify the matching area based on the pivot point. Our proposed algorithm does not need the partitioning to cover all regions and it can produce the more exact matching area information. So our approach is more efficient than the previous DDM matching algorithms and significantly performs the data distribution upon the exact matching area. The remainder of the paper is organized as follows. Section 2 describes HLA issues relevant to data Distribution. The previous algorithms for DDM matching methods are explained in section 3. Section 4 represents the detail proposed algorithm for DDM. Section 5 presents the performance analysis of the system. Finally, section 6 offers conclusion.

II. Overview of DDM in HLA

DDM services, one category of the High Level Architecture (HLA) management reduce message traffic by sending the data only to those federates who need the data for system scalability. The DDM performs filtering based on interest expressions of federates in a federation and allows the federation to control the routing and delivery through user-defined information (i.e., regions). It then distributes objects and interactions from data producers to data consumers. Also, the DDM allows a federate to receive the subscribed attributes after the subscription region is intersected with the publishing federates' update region.

The DDM services allow a value-based filtering. This type of filtering provides the most precise filtering mechanisms which ensure the federates receive the minimal set of data they interest. In the large-scale federation, it is necessary to filter more elaborate for data exchange during the federation execution. Therefore the DDM which provides the value-based filtering can be required for the large-scale simulations with numbers of data exchange. [12]

TABLE I
TERMINOLOGY DEFINITIONS IN THE DDM

Terminology	Definition
Dimension	A named coordinate axis with non-negative integers.
Multidimensional space	A coordinate system whose dimension is d (where d is a fixed natural number)
Range	A continuous semi-open interval on a dimension (lower bound, upper bound)
Region	A set of ranges for any given dimension.
Update region	A specified set of region instance for which is associated by a publishing federate.
Subscription region	A specified set of region instance for which is associated by a subscribing federate.
Overlap	All ranges of dimensions that are contained in the update region and subscription region put one upon another pairwise.
Intersection	An existence when the corresponding region sets overlap.
Matching process	A process to calculate the intersection between update and subscription regions.
Dimension	A named coordinate axis with non-negative integers.
Multidimensional space	A coordinate system whose dimension is d (where d is a fixed natural number)

Table I presents the definitions of terms used in this paper. These definitions originate from the HLA. Although the definitions of terms are provided in Table 1, the terms usage is illustrated and described in Fig. 1. This figure shows an example of region intersection within 2-dimensional space for the value-based filtering. Because our interest lies in the simple exemplification of DDM terms, we assume that the multidimensional space is a 2-dimensional space. In the figure, there are four update regions and four subscription regions. An update region is the defined set of data declared by a publishing federate, whose information is delivered to subscribing federates. A subscription region is the area of interest declared by the subscribing federates. To check the overlapping of update and subscription regions is called intersection. When an intersection exists, data exchange occurs between the publishing federate and the subscribing federate. In this case, there are 2 intersections between update and subscription regions. A process to identify these intersections between update and subscription regions is a matching process. The main role of DDM is to reduce the volume of data exchanged through the matching process during a federation.

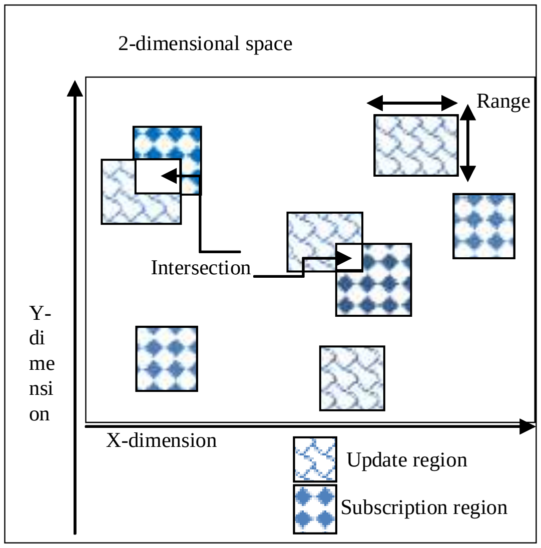


Fig. 1 Example of region intersection in the 2-dimensional space

III. Matching Algorithms in DDM

There are many matching algorithms are appeared for Data Distribution Management of HLA.

A. Region-based Algorithm

The region-based algorithm checks all the pairs of regions until an intersection is found for each pair of update region and subscription regions or the end of the regions list is reached. The implementation of this algorithm is straightforward, but the performance is varies greatly. This approach becomes very efficient when there are many intersections between extents. The main advantage of this algorithm is its simplicity, but it does not scale very well, except when there are many intersections.[1] If there are N update regions and M subscription regions. There are N*M pairs to check in the worst case. However, a considerable amount of computational overhead occurs in the matching process. [11]

B. Grid-based Algorithm

In the grid-based approach, the routing space is partitioned into a grid of cells. Each region is mapped onto these cells. If a subscription region and an update region intersect with the same grid cell, they are assumed to overlap with each other.
[1] In this algorithm, as an exact evaluation of the matching process is not implemented. If update and subscription regions share at least on common grid cell, they are assumed to overlap. Although the overlapping information is not exact, the grid-based algorithm can reduce the computation complexity than the region-based algorithm. [12] The imprecise matching creates irrelevant data communication and additional receiver-side filtering is required. The amount of irrelevant data communicated in the grid-based filtering depends on the grid cell size, but it is hard to define the appropriate size of grid cells. The optimal cell size minimizes the cost of network traffic and updating the group lists associated with each cell. [10]

C. Hybrid Approach

The hybrid approach is an improvement approach over the region-based and the grid-based approaches. The matching cost is lower than the region-based approach, and this advantage is more apparent if the update frequency is high. It also produces a lower number of irrelevant messages than that of the grid-based approach using large cell sizes. [9] The major problem is that it has the same drawbacks as the grid-based approach: the size of the grid cell is very crucial to the behaviour of the algorithm. [1]

D. Sort-based Algorithm

The sort-based algorithm used a sorting algorithm to compute the intersection between update and subscription regions. The intersection is acquired dimension-by-dimension. After processing one dimension, this procedure is repeated for all the other dimensions. The overall overlap information can be obtained by combining the information of each dimension.
[1] However, the sort-based algorithm's performance is degraded when the regions are highly overlapped and it is needed to optimize the sorting data structure for the efficient matching operation. [12]

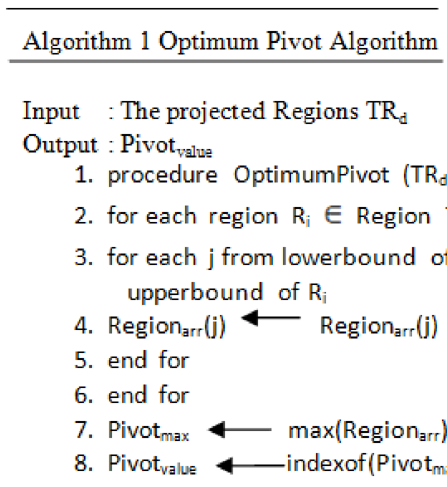
E. Binary Partition-based Algorithm

The binary partition-based matching algorithm takes a divide-and-conquer approach similar to the one used for the quicksort. In Figure 2, this approach consists of two main processes, the repetitive binary partitioning process, and the matching process. First, in the binary partitioning process, the algorithm recursively performs binary partitioning which divides the regions into two partitions that entirely cover those regions. Second, in the matching process, the algorithm uses the concept of an ordered relation which represents the relative location of partition. It easily calculates the intersection between regions on partition boundaries and does not require unnecessary comparisons within regions in different partitions which are located in the ordered relation of partition. The binary partition-based algorithm is not the best choice when the overlapping rate is relatively low. [12]

IV. The Proposed Algorithm

The Layer partition-based matching algorithm is executed in dimension by dimension. This algorithm accepts all regions in the routing space. Then it sends these regions to the Layer partition-based matching algorithm. The final overlapping information can be produced by observing the result of two matrixes for two dimensional routing spaces. The proposed algorithm is completed when all dimensions are covered.

This algorithm works layer by layer for the input regions. The layer partition-based matching algorithm firstly chooses the most suitable pivot based on the region distribution. The pivot point is chosen where the most points are converged. Then it selects the region generally in the first layer into three sets. In the second layer, the specific decision of the regions selection is performed to calculate the matching data between the three sets. This layer also supports the subtracted region lists. This list is subtracted from the input regions set for next matching calculation. The classification of regions is carried out in the region classifier algorithm. The actual matching between the updater regions and the subscriber region are executed. The subtracted region lists is invoked to reduce the next calculation.



```
Algorithm 1 Optimum Pivot Algorithm

Input  : The projected Regions $TR_d$
Output : $Pivot_{value}$
  1. procedure OptimumPivot ($TR_d$)
  2. for each region $R_i \in$ Region $TR_d$ do
  3. for each j from lowerbound of $R_i$ to
        upperbound of $R_i$
  4. $Region_{arr}(j) \longleftarrow Region_{arr}(j)$ ++;
  5. end for
  6. end for
  7. $Pivot_{max} \longleftarrow$ max($Region_{arr}$);
  8. $Pivot_{value} \longleftarrow$ indexof($Pivot_{max}$);
```

Fig. 2 Optimum pivot algorithm

The optimal pivot algorithm selects the most suitable pivot value in the projected regions list. The choice of the pivot value is very important in the divide and conquers approach. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. Some algorithms choose the middle point as the pivot value. The efficiency and performance of the divide and conquer approaches depends on the choice of the pivot value. The optimal pivot algorithm accepts the projected regions list and select one point of that list as the pivot value. At that point, the most subscriber regions and updater regions are converged in the projected regions list. The proposed algorithm can improve the efficiency and performance by the right choice of the optimal pivot algorithm. In figure 2, the optimal pivot algorithm decides the optimal pivot value instead of the middle point.

The first layer partition algorithm generally defines the projected regions list into three sets such as the left set, the right set and the pivot set. It keeps the left set which possess the less upper bound of the regions than the pivot value. The right set holds the greater lower bound of the regions than the pivot value. It keeps the pivot set including the regions along the pivot point. During the creating of the pivot set, the first layer partition algorithm looks for the maximum upper bound and minimum lower bound of the pivot set. This fact is very important to apply the projected regions at some step of the second layer partition algorithm.

The second layer partition algorithm computes the specific region sets to employ in the intersection calculation steps. It chooses the matched regions from the general left and right sets according to the results of the previous steps, the first layer partition algorithm. The minimum lower bound is used to determine the left match set. The maximum upper bound is used to find out the right match set. The left subtract set is obtained by comparing the lower bound of the left set's regions with the minimum lower bound value. The upper bound of the right set's regions are evaluated by the maximum upper bound to find out the right subtract set. Both subtract sets are taken away in the next repetition of the proposed algorithm.

The region classifier algorithm accepts the regions set. It organizes the two region sets. The one is subscriber set and another one is the updater set. All of the regions in the input set are classified and put to the specific kinds of region set.

The intersection calculation algorithm keeps the overlapping result between the input subscriber set and the input updater set. The result is recorded in the two dimensional matrix. The proposed algorithm calculates the overlapping result repeatedly. The system flow is represented in figure 4.

Algorithm 2 Layer Partition-based Matching Algorithm

```
Input: Region $TR_d$
Output : n-by-n matrix, OM = ($om_{ij}$), where i, j $\in$ n
1. procedure LayerPartitionbasedMatching (Region $TR_d$)
2. Set a Pivot value P, OptimumPivot($TR_d$);
3. Use the FirstLayerPartition ($TR_d$, P) to find three sets and two values $S_{left}$, $S_{pivot}$, $S_{right}$, $max_{UB}$, $min_{LB}$;
4. Set LeftMatchSet and $S_{subl}$, SecondLayerPartition ($S_{left}$, P, $min_{LB}$);
5. Set RightMatchSet and $S_{subr}$, SecondLayerPartition ($S_{right}$, P, $max_{UB}$);
6. Set $S_{ps}$ and $S_{pu}$ of Pivot, RegionClassifier ($S_{pivot}$);
7. Set $S_{ls}$ and $S_{lu}$ of LeftMatchSet, RegionClassifier (LeftMatchSet);
8. Set $S_{rs}$ and $S_{ru}$ of RightMatchSet, RegionClassifier (RightMatchSet);
9. Set $LSub_s$ and $LSub_u$ of $S_{subl}$, RegionClassifier ($S_{subl}$);
10. Set $RSub_s$ and $RSub_u$ of $S_{subr}$, RegionClassifier ($S_{subr}$);
11. for each region $R_i \in S_{pu}$ do
12. for each region $R_j \in S_{ps}$ do
13.   $om_{ij}$++;
14. end for
15. end for
16. IntersectionCalculation ($S_{ls}$, $S_{pu}$);
17. IntersectionCalculation ($S_{lu}$, $S_{ps}$);
18. IntersectionCalculation ($S_{pu}$, $S_{rs}$);
19. IntersectionCalculation ($S_{ps}$, $S_{ru}$);
20. IntersectionCalculation ($LSub_s$, $LSub_u$);
21. IntersectionCalculation ($RSub_s$, $RSub_u$);
22. $S_{ls} \longleftarrow S_{ls} - LSub_s$;
23. $S_{lu} \longleftarrow S_{lu} - LSub_u$;
24. $S_{rs} \longleftarrow S_{rs} - RSub_s$;
25. $S_{ru} \longleftarrow S_{ru} - RSub_u$;
26. IntersectionCalculation ($S_{ls}$, $LSub_u$);
27. IntersectionCalculation ($S_{lu}$, $LSub_s$);
28. IntersectionCalculation ($RSub_u$, $S_{rs}$);
29. IntersectionCalculation ($RSub_s$, $S_{ru}$);
30. $S_{left} \longleftarrow S_{ls} \cup S_{lu}$;
31. $S_{right} \longleftarrow S_{rs} \cup S_{ru}$;
32. LayerPartitionbasedMatching (Region $S_{left}$) ;
33. LayerPartitionbasedMatching (Region $S_{right}$) ;
```

Fig. 3 Layer partition-based matching algorithm

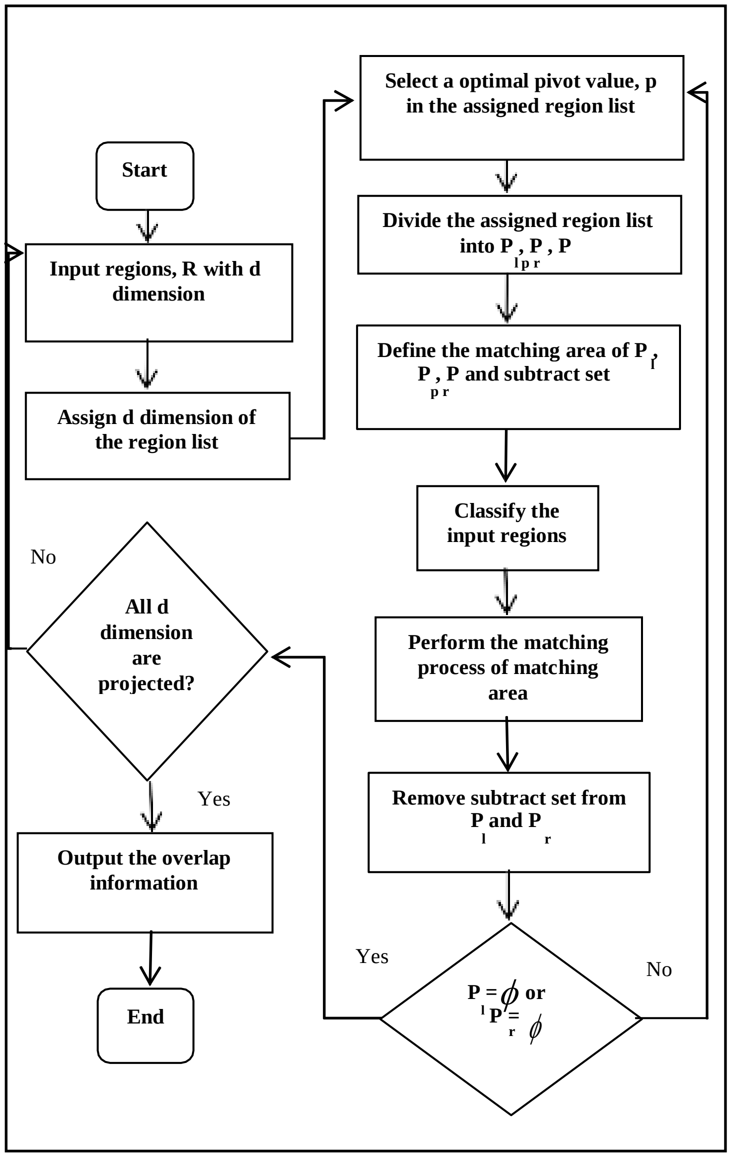

Start
Input regions, R with d dimension
Assign d dimension of the region list
Select a optimal pivot value, p in the assigned region list
Divide the assigned region list into P_l, P_p, P_r
Define the matching area of P_l, P_p, P_r and subtract set
Classify the input regions
Perform the matching process of matching area
Remove subtract set from P_l and P_r
P_l = ϕ or P_r = ϕ
Yes
No
All d dimension are projected?
No
Yes
Output the overlap information
End

Fig. 4 System flow of the proposed algorithm

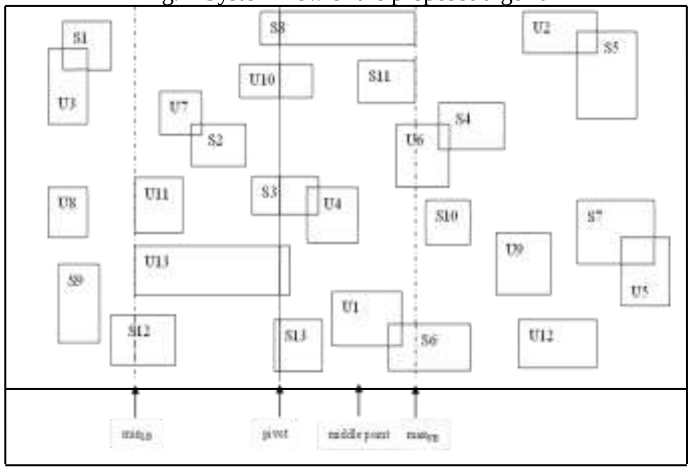


Fig. 5 Sample routing space with 13 updaters and 13 subscribers

V. The Performance Analysis

This section presents an experimental evaluation of the proposed method. We implemented the algorithm with Java language and processed in the HLA-based distributed simulations. We assume that our experimental environment has a 2-dimensional space. As the performance of the DDM execution time for the matching process is measured, our experiments were conducted using Microsoft Windows 7 with a 2.0GHz Intel(R) Core(TM) i7 CPU and 8GB memory. One of the important experimental parameter is the number of regions. This parameter is capable of characterizing the entire situation of regions for the matching process. It represents the scalability of the system. In the matching process, the regions are distributed randomly across the space. Finally, the overlap rate is defined as the proportion of the scene volume occupied by the regions.

$$\text{overlap rat} = \frac{\sum \text{area of regions}}{\text{area of space}}$$

If the space is 100 * 100 and one region is 1 * 1, where the number of region is fixed at 100, the overlap rate is

$$0.01 = \frac{100 * (1*1)}{100 * 100}$$

The performance of the proposed algorithm is presented in the following table. The experimental result is calculated based on the overlapping degree. We use the space is 1000* 1000 and the execution time of matching algorithm is calculated according to three overlapping degree. They are 0.01, 0.1 and 1 overlapping degree. We distributed the regions randomly over the routing space. We experiment the execution time of the proposed algorithm upon different numbers of region from 1000 to 10000.

The proposed algorithm is best when the overlapping degree is 1 upon every number of input regions. The result is opposing the previous matching algorithms. When the overlapping degree is low, also the size of the input regions is small; the execution time is very high. Now the same size of all input region is used at each number of regions group. The proposed algorithm is mainly search the optimal pivot selection and defining the matching area and subtracted regions list. Those data is depending on the size of region. The proposed algorithm can be used efficiently among the different size of regions in each number of regions group. The greater the overlapping degree is, the better the performance of the proposed algorithm.

TABLE II
PERFORMANCE ANALYSIS FOR MATCHING

No. of Regions	Execution time (≈ seconds) overlap rate 1	Execution time (≈ seconds) overlap rate 0.1	Execution time (≈ seconds) overlap rate 0.01
1000	0.28	0.83	1.53
2000	1.10	2.88	7.83
3000	2.60	7.66	26.78
4000	4.93	15.53	47.76
5000	7.84	23.46	71.76
6000	12.34	41.85	124.56
7000	17.54	54.60	146.32
8000	22.43	71.17	170.45
9000	32.5	93.39	192.87
10000	49.60	132.94	214.88

VI. CONCLUSION

Efficient data distribution is an important issue in large scale distributed simulations. The broadcasting mechanism employed in Distributed Interactive Simulation (DIS) standards generates unnecessary network traffic and is unsuitable for large scale and dynamic simulations. In this paper interest new region detection algorithm in particular. The proposed system also does not include the filtering cost of the irrelevant messages by using layer partition-based algorithm. The platform can also be easily modified to support other matching algorithm and the experimental results can then be used to compare the efficiencies of the various matching mechanisms.

ACKNOWLEDGMENT

I firstly thank my parent for their support in my pursuit of the research. I especially thank and appreciate to my teachers for allowing me to develop this research. Heartfelt thanks are extended to my supervisor, **Dr Nay Min Tun**, Associate Professor, Principal of Computer University, Kyaing Tong, for his invaluable recommendations, detailed guidance, patient supervision and keen interest throughout the preparation of this research.

REFERENCES

[1] C. Raczy, G. Tan, J. Yu. "A Sort-Based DDM Matching Algorithm for HLA", *ACM Transactions on Modeling and Computer Simulation (TOMACS*), Vol. 15 Issue 1, 2005.

[2] M. T. Nwe Nwe, T. Nay Min, "Optimization of Region Distribution Using Binary Partition-based Matching Algorithm for Data Distribution Management", *International Journal of Engineering Research & Technology (IJERT)* Vol.2 Issue 2, Feb. 2013

[3] J. S. Steinman, K. Morse. "Data Distribution Management in HLA: Multidimensional Regions and Physically Correct Filtering". Proc. Spring Simulation Interoperability Workshop, 1997.

[4] J. S. Damann, R. M. Fujimoto and R. M. Weatherly. "The DoD High Level Architecutre: An Update", Proc. Simulation Conference, Dec. 1998.

[5] I. Tacic, R. T. Fujimoto. "Synchronized Data Distribution Management in Distributed Simulations", Proc. 12th Workshop on Parallel and Distributed Simulation, 1998.

[6] K. L. Morse, K. Tsai, L. Bic, "Multicast Grouping for Dynamic Data Distribution Management", Proc. Summer Computer Simulation Conference, cs.bham.ac.uk, 1999.

[7] A. Boukerche and A. Roy. "In Search of Data Distribution Management in Large Scale Distributed Simulations", Proc. Summer Simulation Conference, 2000.

[8] A. Boukerche, A. Roy, and N. Thomas. "Dynamic Grid-Based Multicast Group Assignment in Data Distribution Management". Proc. 4th International Workshop on Distributed Simulation and Real-Time Applications, 2000, p 47–54.

[9] G. Tan, R. Ayani, and Y. Zhang. "A Hybrid Approach to Data Distribution Management". Proc. 4th International Workshop on Distributed Simulation and Real-Time Applications, 2000, p. 55–61.

[10]R. Ayani, F. Moradi and G. Tan. "Optimizing cell-size in grid-based DDM", Proc. 14th Workshop on Parallel and Distributed Simulation, Bologna Italy, May, 2000, p. 93–100.

[11]Y. Jun, C. Raczy and G. Tan. "Evaluation of sort-based matching algorithm for the DDM", Proc. 16th Workshop on Parallel and Distributed Simulation, Washington DC, May. 2002, p. 68–75.

[12]C. Sung, J. Ahn, T. G. Kim. "A Binary Partition-Based Matching Algorithm for Data Distribution Management", Proc. Winter Simulation Conference, 2011.