ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Layer Partition-based Matching Algorithm of DDM based on Dimension

Nwe Nwe Myint Thein[1], Nay Min Tun[2]

[1]Ph.D student, University of Technology (Yatanarpon Cyber City), Myanmar,
[2]Associate Professor, Computer University (Kyaing Tong), Myanmar

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 compares the performance of the layer partition-based matching algorithm based on the result of previous calculated dimension. The algorithm chooses the dynamic pivot based on regions distribution in the routing space. 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 firstly calculates the X-dimension for matching result. Then modify the input region of Y-dimension based on the matching result of the previous dimension. The proposed algorithm is more efficient most of overlapping degree but sometime the proposed algorithm has more execution time than the original layer partition-based algorithm.

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. 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 compare a layer partition-based algorithm with the update layer partition-based matching algorithm. The update algorithm is based on the matching result of the

previous dimension. 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 compares the performance analysis of the system based on the dimension. Finally, section 6 offers conclusion.

II. OVERVIEW OF DDM IN HLA

Within the Department of Defence (DoD), the HLA provides a common architecture for modelling and simulation. HLA's Run Time Infrastructure (RTI) provides commonly required services to simulation systems that service a distributed operating system providing to applications. HLA RTI's Data Distribution Management (DDM) mechanisms are necessary to provide efficient scalable support for large scale distributed simulations.

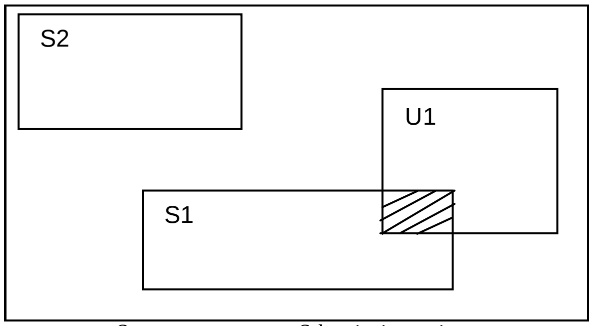
The fundamental Data Distribution Management constructs a routing space. A routing space is a named sequence of dimensions, which form a multi-dimensional coordinate system. Regions are defined as sets of extents which are rectangles representing the sensor ranges of different units. Federates either express an interest in receiving data (subscribe) or declare their intention to send data (publish). These intentions are expressed through:

Subscription Region: Bounding routing space coordinates which narrow the scope of interest of the subscription federate.

Update Region: Bounding routing space coordinates which are guaranteed to enclose an object's location in the routing space. Both subscription and update regions can change in size and location over time as a federate's interests change or an object's location in the routing space changes.

An object is discovered by a federate when at least one of the object's attributes come into scope for federate, i.e. if an only if: the federate has subscribed to the attribute of the object's update region overlaps the federate's subscription region. At a time step, that federate will ensure the characteristics of the object instance which map to the dimensions of the routing space falling within the extents of the associated region there by the attribute update is being issued. As the state of the objects change, the federate may need to either adjust the extents on the associated regions or change the association to another region. Each federate can create multiple update and subscription regions. Update regions are associated with individual objects that have been registered with the RTI. A federate might have a subscription region for each sensor system being simulated.[5]

Figure 1 show on update region (U1) and two subscription regions (S1, S2) within a two dimensional routing space. U1 and S1 overlap and therefore attributes and interactions associated with U1 will be routed by the RTI to the federate that created S1. On contrast U1 and S2 do not overlap and attributes and interactions will not be routed from the federate that create U1 to the federate that created S2.



S	--> Subscription region
U	-->Update region
Shaded area	-->Overlap region

Figure 1. Two-dimensional routing space

III. MATCHING ALGORITHMS IN DDM

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. 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. First, construct an end points list of all regions for each dimension of the routing space. Each list contains the coordinates of all the extents of the associated dimension. Second, sort all the lists according to their coordinates. Finally, scan each sorted list from the left side to obtain the overlap information of each dimension. The sort-based algorithm maintains the set of subscription regions before and after the current position. Therefore, it is possible to know exactly, for each update region, which subscription regions match on 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. [11]

D. 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. The system flow is represented in figure 2.

This algorithm works layer by layer for the input regions. The optimal pivot algorithm selects the most suitable pivot value in the projected regions list. 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. 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. 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. [13, 14]

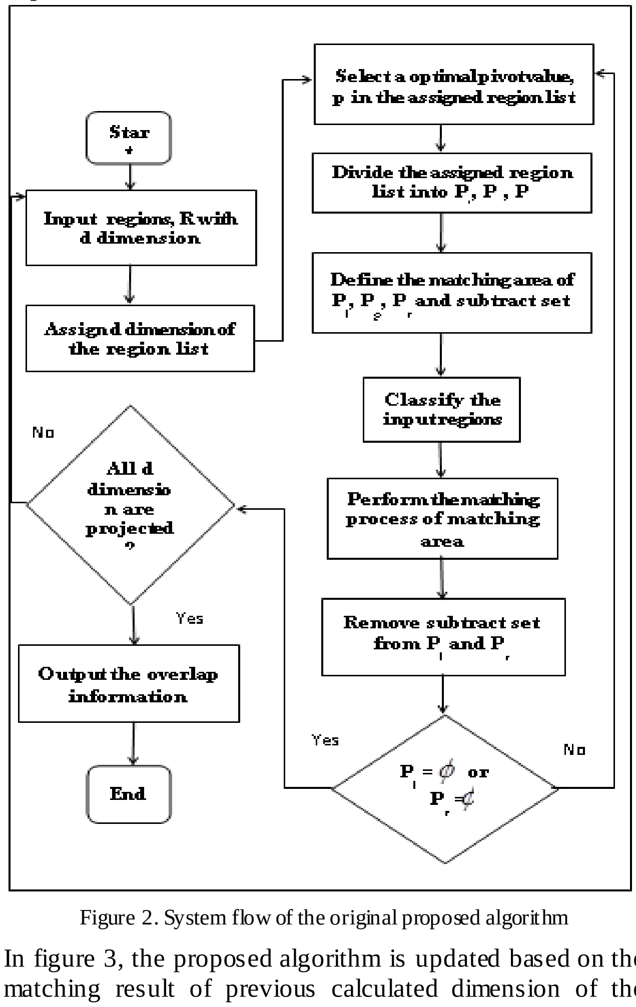

Star t
Input regions, R with d dimension
Assign d dimension of the region list
No
All d dimensio n are projected ?
Yes
Output the overlap information
End
Select a optimal pivot value, p in the assigned region list
Divide the assigned region list into P_l, P_c, P_r
Define the matching area of P_l, P_c, 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
Yes
No
P_l = ϕ or P_r = ϕ

Figure 2. System flow of the original proposed algorithm

In figure 3, the proposed algorithm is updated based on the matching result of previous calculated dimension of the routing space. The layer partition-based matching algorithm firstly calculates the X-dimension of the input regions list. Before calculating the Y-dimension, the input region list of the Y-dimension is updated and reduced. The new input regions list is calculated according to the overlapping result of the updater regions and subscriber regions of the X-dimension. The algorithm detects the non-overlapping result of the regions in the X-dimension. Then the updated layer partition-based matching algorithm defines the new input regions list of the Y-dimension. The detail processes of the updated layer partition-based matching algorithm is shown the following figure.

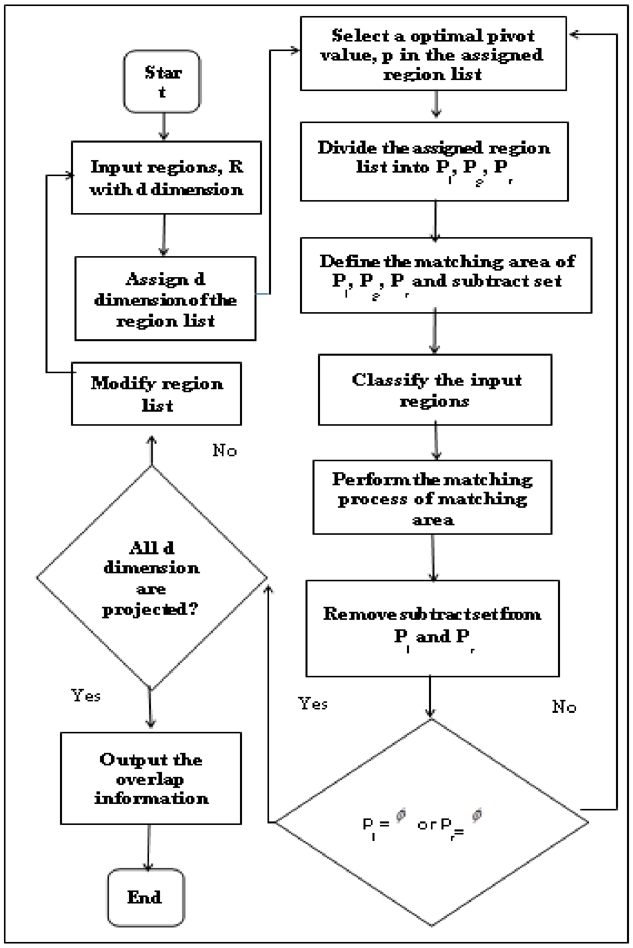

Star t
Input regions, R with d dimension
Assign d dimension of the region list
Modify region list
No
All d dimension are projected?
Yes
Output the overlap information
End
Select a optimal pivot value, p in the assigned region list
Divide the assigned region list into P_l, P_g, P_r
Define the matching area of P_l, P_g, 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
Yes
No
$P_l = \emptyset$ or $P_r = \emptyset$

Figure 3. System flow of the proposed algorithm based on dimension

V. REFERENCES

The original layer partition-based matching 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 1. Analysis of the original algorithm

Sr. No.	*No. of regions*	*Execution time (≈ seconds) overlap rate 1*	*Execution time (≈ seconds) overlap rate 0.1*	*Execution time (≈ seconds) overlap rate 0.01*
1	1000	0.28	0.83	1.53
2	2000	1.10	2.88	7.83
3	3000	2.60	7.66	26.78
4	4000	4.93	15.53	47.76
5	5000	7.84	23.46	71.76
6	6000	12.34	41.85	124.56
7	7000	17.54	54.60	146.32
8	8000	22.43	71.17	170.45
9	9000	32.5	93.39	192.87
10	10000	49.60	132.94	214.88

Table 2. Analysis of the update algorithm

Sr. No.	*No. of regions*	*Execution time (≈ seconds) overlap rate 1*	*Execution time (≈ seconds) overlap rate 0.1*	*Execution time (≈ seconds) overlap rate 0.01*
1	1000	0.27	0.74	0.87
2	2000	1.02	**5.05**	7.36
3	3000	2.51	6.51	**27.94**
4	4000	4.31	13.74	**48.65**
5	5000	7.22	21.41	**76.85**
6	6000	11.47	35.78	**124.97**
7	7000	16.92	48.065	**146.62**
8	8000	20.67	64.33	**171.3**
9	9000	29.93	82.27	**194.06**
10	10000	45.81	118.09	**216**

The update algorithm is more efficient on overlapping all regions in degree 1 and 0.1 except 2000 regions. The reduce execution time is large in overlapping degree 0.1.

The execution time of overlapping degree 0.01 is not good except 1000, 2000 and 3000 regions. Therefore the layer partition-based matching algorithm depends on the size of the input regions. The larger difference between the regions' size is more efficient.

VI. CONCLUSION

Efficient data distribution is an important issue in large scale distributed simulations with several thousands of entities. 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 present the performance of the layer partition-based matching algorithm based on the previous calculated dimension. The proposed algorithm also provides the minimum matching cost between the updaters and the subscribers within the routing space. The experimental results can then be used to compare the efficiencies of the various matching mechanisms.

IV. 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.

[13] M. T. Nwe Nwe, T. Nay Min, "Dynamic Pivot for Layer Partition-based Matching Algorithm of DDM based on Regions Distribution", Proc. Fourth International Conference on Science and Engineering (ICSE 2013), Yangon, Myanmar, December, 2013.

[14] M. T. Nwe Nwe, T. Nay Min, "Layer Partition-based Matching Algorithm of DDM", Proc. 3rd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2014), Singapore, February, 2014.