

Optimization of Region Distribution Using Binary Partition-based Matching Algorithm for Data Distribution Management

Nwe Nwe Myint Thein, Nay Min Tun

University of Technology (Yadanarpon Cyber City)

Abstract

High Level Architecture (HLA) is an 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. 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 a region distribution detection algorithm to use in the binary partition-based matching algorithm for DDM in the HLA-based large-scale distributed simulations. The new region detection algorithm calculates the distribution of regions in the routing space after performing the partitioning of regions that contains in the projected routing space. The partitioning among regions performs based on the result of optimal pivot. Then, the proposed algorithm calculates the regions distribution and define the matching area of the selected partitions in routing space. 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 and reduces the irrelevant message among federates.

1. Introduction

The High Level Architecture (HLA) is an 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's DDM service is one of the most filtering mechanisms regarding large scale distributed simulations. [2] 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 region distribution detection 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 intends to ease the irrelevant message routing over the network. This algorithm firstly calculate the optimal pivot value to divide the routing space. Then it detects the position of regions and calculate the region distribution status according to their coordinates. Then the proposed algorithm need to specified the matching based on the partitioning point. Our proposed algorithm can produce the more exact matching information in a short time. So our approach is more efficient than the previous DDM matching algorithms and significantly perform 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 is explained in section 3. Section 4 represents the

detail proposed algorithm for DDM. Finally, section 5 offers conclusion and further work.

2. Overview of DDM in HLA

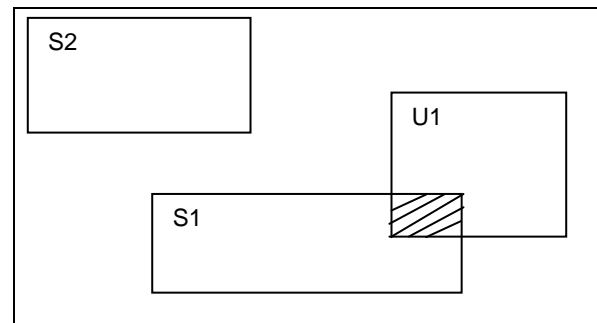
Within the Department of Defense (DoD), the HLA provides a common architecture for modeling and simulation. HLA's Run Time Infrastructure (RTI) provides commonly required services to simulation systems that service a distributed operating system providing to applications. HLA RTI's Data Distribution 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]



S --> Subscription region

U --> Update region

Shaded area --> Overlap region

Figure 1. Two-dimensional routing space

Figure 1 shows an 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.

3. Matching algorithms in DDM

3.1. 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 varies greatly. For figure 1, the region-based filtering S1 receives all updates within U1. S1 receives some irrelevant data (those within U1, but outside of the shaded area) as well. 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]

3.2. Grid-based algorithm

In the grid-based approach, the routing space are partitioned into a grid of cells. Each regions 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.[11] 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. In 2000, the researchers have shown that the grid cell size has a substantial impact on the network traffic and affects the performance of distributed simulations. They have formulated the calculation of the optimal grid cell size for a time-stepped distributed simulation scenario. The optimal cell size minimizes the cost of network traffic and updating the group lists associated with each cell. [9]

3.3. 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. [8] 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]

3.4. 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 maintain 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]

3.5. Binary partition-based algorithm

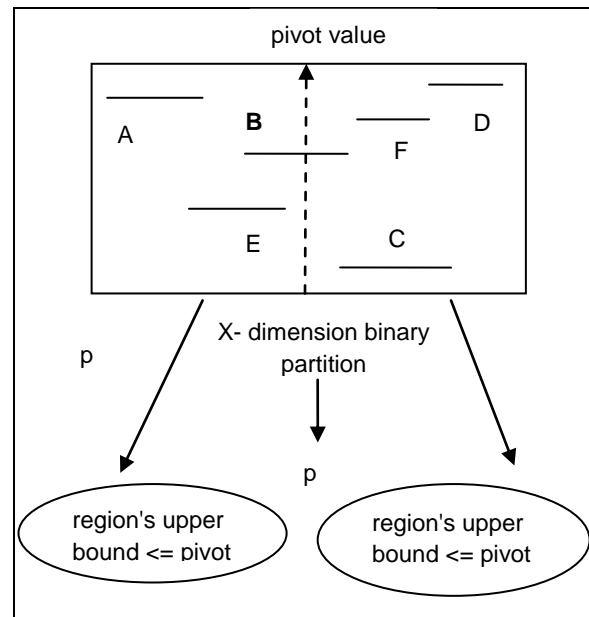


Figure 2: Binary partition into three partitions, p_{left} , p_{right} and p_{pivot}

The binary partition-based matching algorithm takes a divide-and-conquer approach similar to the one used for the quicksort. In Figure 3, 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. [11]

4. The proposed algorithm

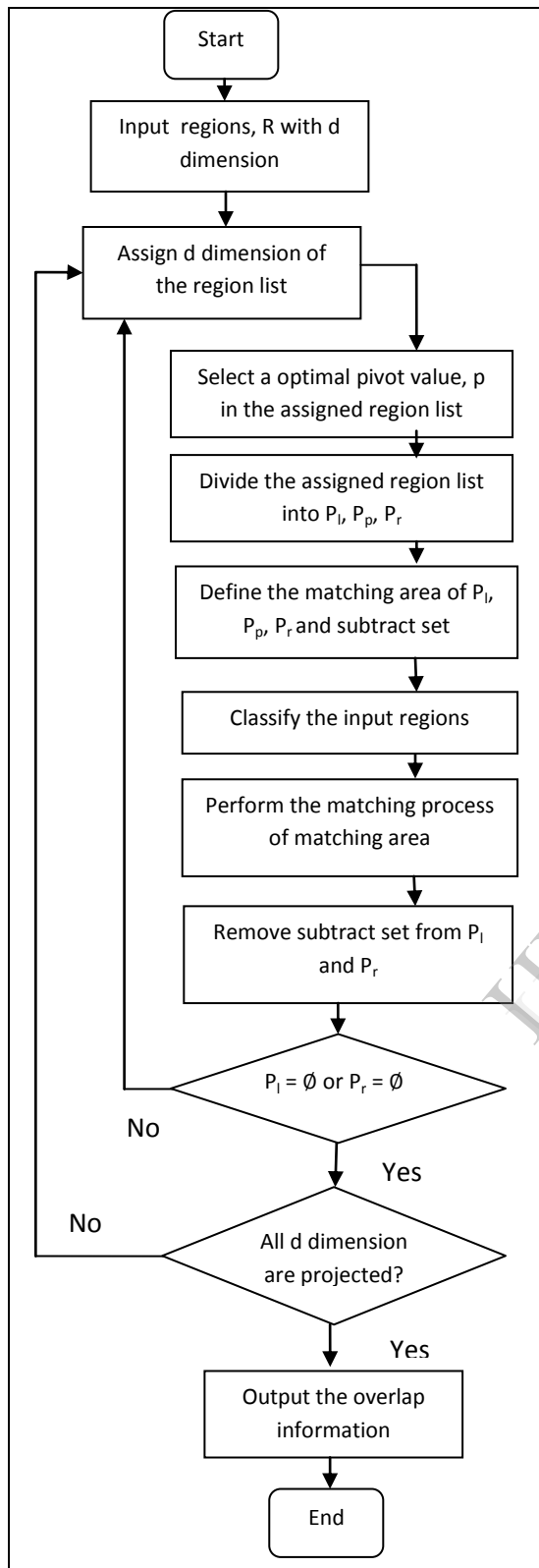


Figure 3: The steps of the proposed system

The proposed algorithm calculate the intersection information between the update regions and subscription regions. This algorithm based on the concept of the binary partition-based matching

algorithm. The proposed algorithm include six main processes. Firstly, the proposed algorithm chooses one point of the routing space as the pivot value of the routing space where converge a lot of regions. The second process divide the routing space into three sets based on the calculated pivot value. Each set include the update regions and subscription regions values. They are pivot set , left set and right set of the pivot value. The matching area (MA) partition is defined in the third process. The proposed algorithm intends to match the update regions and subscription regions of the pivot set with the update regions and subscription regions within the matching area.

The proposed algorithm do not need to match all regions of the left set and right set with the pivot set. In this stage, the proposed algorithm collects the subtract set of the projected set. These sets are need to take from the left and right set during the next repetition of the proposed algorithm. Before the execution of the intersection calculation, all collected sets need to classify which regions are updaters and which are subscribers. Finally, the intersection calculation produces the matching result between the update regions and subscription regions of the defined matching area. The proposed algorithm continues the next execution after the subtraction process from the left and right regions sets.

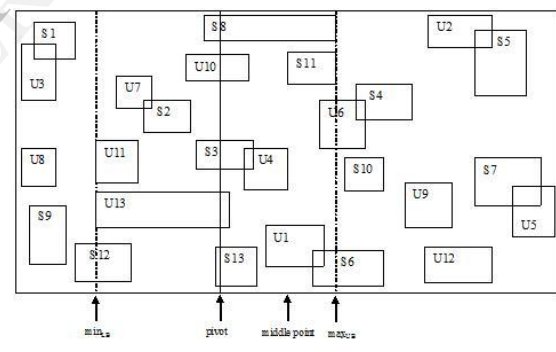


Figure4: The sample routing space

In the sample routing space, the optimum pivot algorithm choose the most suitable pivot value where the five regions are converged. Then the first layer partition algorithm divide the routing space into three sets. The left set of the pivot value includes {S1, S2, S9, S12, U3, U7, U8, U11} regions. The regions {S4, S5, S6, S7, S10, S11, U1, U2, U4, U5, U6, U9, U12} are contained in the right set of the pivot value. The pivot set consists of {S3, S8, S13, U10, U13} regions. This algorithm also calculates the minimum lower bound and the maximum upper bound of the pivot set. The lower bound of U13 is the minimum lower bound and the

upper bound of S8 is the maximum upper bound. It is also known as defining the matching area of the projected region list. The second layer partition algorithm defines the matching sets for the left set and right set including the subtract region set to subtract the regions in the next repetition of the proposed algorithm. This algorithm produces the result of the matching sets into two region sets. They are {S2, S12, U7, U11} for left match set and {S6, S11, U1, U4, U6} for right match set. The left match set of {S2, U7, U11} and the right match set of {S11, U1, U4} are created to subtract from the left set and right set in the next repetition of the proposed system. All of these sets need to classify which region is updater and which region is subscriber. So the region classifier algorithm supports these procedure. It categorizes the regions of all sets. The intersection calculation algorithm is invoked when the two match sets are compared with the pivot set. After finishing the overlapping information, the left and right subtract sets are taken from the left set and the right set. It also performs the same operation from the two match set. The two subtract sets' regions also need to check the intersection information with the remaining regions of the two match sets. After the first execution of the proposed algorithm, the following overlapping information can be obtained. For the pivot set regions, U10 and U13 are overlapped with the same subscriber regions S3, S8 and S13. U10 with S2, U13 with S2 and S12, U7 with S2 and S12 with U7 and U11 are the overlapping information for the left match set. The overlapping results for the right match are S3 with U4, S8 with U1, U4 and U6, S13 with U4, S11 with U1 and U4, S6 with U1 and U4 and U6 with S11. The proposed algorithm performs dimension by dimension. It projects the regions of the routing space and calculates the overlapping information by using the above algorithms. The proposed algorithm continues the next repetition until all regions are covered.

5. 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 interest new region detection algorithm in particular. A simulation platform implement data distributed management in detecting enemy's aircraft from navy's warship by using new region detection algorithm. This platform serves as a convenient tool for implementing the new region detection algorithm in data distribution management. The proposed system can detect the aircraft's status from warships in an efficient and timely manner. The system also does not include the filtering cost of the irrelevant messages. The proposed algorithm also provide the minimum matching cost between the updaters (the aircrafts) and the subscribers (the warships) within the routing space. 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.

6. References

- [1] C. Racz, G. Tan, J. Yu. "A Sort-Based DDM Matching Algorithm for HLA", *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, Volume 15 Issue 1, 2005 .
- [2] J. S. Steinman , K. Morse. "Data Distribution Management in HLA: Multidimensional Regions and Physically Correct Filtering". Proceedings of the Spring Simulation Interoperability Workshop, 1997.
- [3] J. S. Damann, R. M. Fujimoto and R. M. Weatherly. "The DoD High Level Architecture: An Update", Proceedings of the Simulation Conference Proceedings, 1998 December.
- [4] I. Tadic, R. T. Fujimoto . "Synchronized Data Distribution Management in Distributed Simulations", Proceedings of the 12th Workshop on Parallel and Distributed Simulation, 1998.
- [5] K. L. Morse, K. Tsai, L. Bic, "Multicast Grouping for Dynamic Data Distribution Management", Proceedings of the Summer Computer Simulation Conference, cs.bham.ac.uk, 1999.
- [6] A. Boukerche and A. Roy. " In Search of Data Distribution Management in Large Scale Distributed Simulations", Proceedings of the Summer Simulation Conference, 2000.
- [7] A. Boukerche, A. Roy, and N. Thomas. "Dynamic Grid-Based Multicast Group Assignment in Data Distribution Management". Proceedings of the 4th International Workshop on Distributed Simulation and Real-Time Applications, 2000, pp 47-54.

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

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

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

[11] C. Sung, J. Ahn, T. G. Kim. "A Binary Partition-Based Matching Algorithm For Data Distribution Management", Proceedings of the Winter Simulation Conference, 2011.

IJERT