Performance Analysis of Layer Partition-based Matching Algorithm in Data Distribution Management

Nwe Nwe Myint Thein
University of Information Technology, Myanmar
nwenwemyintthein@uit.edu.mm

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. The evolution of the DDM service in HLA provides solutions to problems by using filtering mechanism that is suitable for large-scale simulation. 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. The paper analyzes the performance of layer partition-based algorithms (LPM) for the matching process based on the different overlapping rate. The LPM algorithm provides the more definite matching area between update region and subscription region. The LPM algorithm guarantees low computational overheads for matching process of higher overlapping rate between the regions and reduce the irrelevant message among federates.

Keywords- HLA, DDM, Layer Partition-based, Matching Algorithm, Modeling and Simulation, Distributed System

1. Introduction

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.

DDM is a set of services defined in HLA to distribute information in distributed simulation environments. 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 RTI to coordinate the operations and the exchanges of data between federates (simulations) during a runtime execution. The interaction of object instances support by the function of RTI, which is similar to a distributed operating system. A simulation platform implements data distributed management in war game, airport modeling and simulation, air traffic control system and public transportation domain.

The remainder of the paper organizes as follows. Section 2 describes HLA issues relevant to data distribution. The previous algorithms for DDM matching methods explain in section 3. Section 4 represents the LPM algorithm for DDM. Section 5 presents the performance analysis of the system. Finally, section 6 offers conclusion.

2. Overview of DDM in HLA

DDM utilizes an N-dimensional coordinate system called a routing space to represent, for example, a geographical area. Federates express their interest by defining subscription regions that characterize the information they are interested in receiving. Each message is associated with a publication region to characterize the content of the message. If an overlap detect between a message publication region and a subscription region, the message will send to that subscribing federate. The main role of DDM is to reduce the volume of data exchanged through the matching process during a federation. Figure 1 shows the sample 2-deimensional routing space with four subscription regions and five publication (Update) regions.

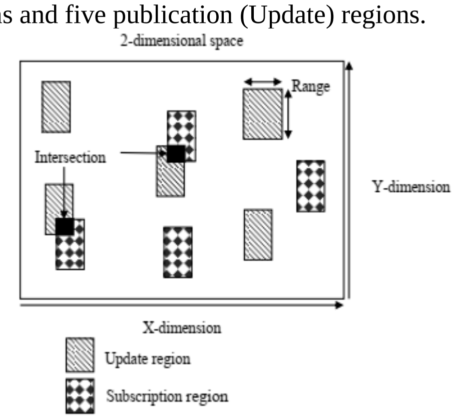

Figure 1: Example of region intersection in the 2-dimensional space

Table 1 presents the definitions of terms used in this paper, which originated from the HLA. [7]

ISBN 978-99971-0-585-1

Table 1: Terminology Definition 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.

3. Matching Algorithms in DDM

3.1. Region-based Algorithm

The region-based algorithm checks all the pairs of regions until an intersection found for each pair of update region and subscription regions or the end of the regions list reached. The implementation of this algorithm is straightforward, but the performance is varying greatly. [5] If there is N update regions and M subscription regions. “There are N*M pairs to check in the worst case. [7]”

3.2. Grid-based Algorithm

In the grid-based approach, the routing space partition into a grid of cells. Each region mapped onto these cells. If a subscription region and an update region intersect with the same grid cell, they assumed to overlap with each other. [9] Although the overlapping information is not exact, the grid-based algorithm can reduce the computation complexity than the region-based algorithm. [8] 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. [7]

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. [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 behavior of the algorithm. [6]

3.4. Sort-based Algorithm

The sort-based algorithm used a sorting algorithm to compute the intersection between update and subscription regions. [5] However, the sort-based algorithm's performance degraded when the regions are highly overlapped and it needed to optimize the sorting data structure for the efficient matching operation. [10]

3.5. Binary Partition-based Algorithm

The binary partition-based matching algorithm takes a divide-and-conquer approach similar to the one used for the quicksort.

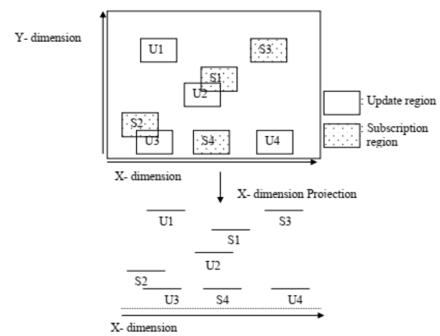

Figure 2: Dimension Projection with the X Dimension

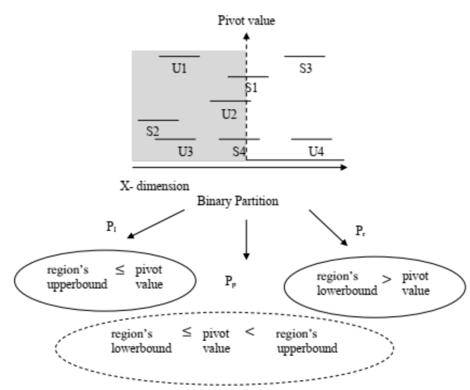

Figure 3: Binary Partition into Three Partitions, Pl, Pp and Pr

This approach consists of two main processes, the repetitive binary partitioning process, and the matching process. In the binary partitioning process, the algorithm recursively 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 process of algorithm describes in figure 2 and figure 3. The binary partition-based algorithm is not the best choice when the overlapping rate is relatively low. [7]

4. Layer Partition-based Matching Algorithm (LPM)

The Layer partition-based matching algorithm supports to search the overlapping information for data distribution management of HLA. It executes in dimension by dimension. This algorithm accepts all regions in the routing space. It also generates all regions randomly. Then it sends these regions to the Layer partition-based matching algorithm. The final overlapping information produces by observing the result of two matrixes for two-dimensional routing spaces. The LPM algorithm complete when all dimensions are covered. The detail instruction is state in figure 4. [3]

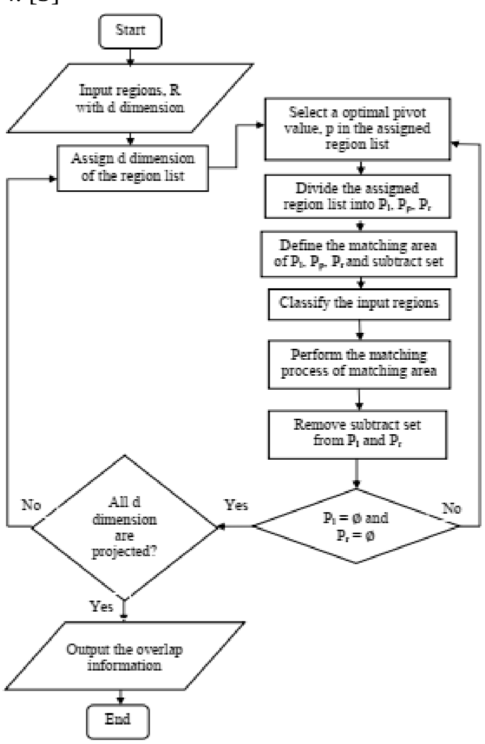

Figure 4: The LPM Dimension Algorithm

The LPM algorithm firstly chooses the optimal pivot to define the matching area. The efficiency and performance of the divide and conquer approaches depends on the choice of the pivot value. Some algorithms choose the middle point as the pivot value. The 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. In figure 5, the optimal pivot algorithm decides the optimal pivot value instead of the middle point. [2]

To define the exact matching area, a region distribution detection algorithm mainly used in the first layer of layer partition-based matching algorithm. The LPM 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 algorithm guarantees low computational overheads for matching process and reduces the irrelevant message among federates. [1]

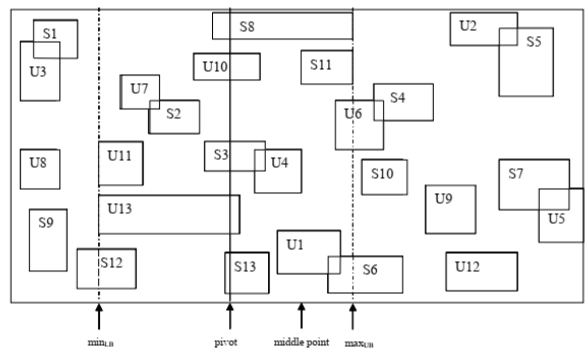

Figure 5: Sample Routing Space with 13 updaters and 13 subscribers

Table 2: Performance Analysis for Pivot Choosing

No. of updaters	No. of subscribers	% of overlap region	No. pivot choosing	
			Binary Partition-based Algo	Layer Partition-based Algo
13	13	0	32	15
13	13	7	32	15
13	13	15	32	15
13	13	23	32	15
13	13	30	32	15
13	13	38	32	15
13	13	46	32	15
13	13	53	32	15
13	13	61	32	15
13	13	69	32	15
13	13	76	32	15
13	13	84	31	16
13	13	92	31	16
13	13	100	30	16

Table 3: Performance Analysis for Matching

No. of updaters	No. of subscribers	% of overlap region	No. pivot choosing	
			Binary Partition-based Algo	Layer Partition-based Algo
13	13	0	44	20
13	13	7	44	20
13	13	15	44	21
13	13	23	43	20
13	13	30	43	22
13	13	38	43	20
13	13	46	43	20
13	13	53	44	22
13	13	61	43	22
13	13	69	42	20
13	13	76	42	22
13	13	84	42	21
13	13	92	42	21
13	13	100	43	21

The LPM algorithm promises the lower number of pivot point choosing. It also reduces the number of matching process between the updater regions and the subscriber regions of the routing space. The analysis of the LPM with 13 updaters and 13 subscribers describe in Table 2 and Table 3. The area of the routing space is 100*60. We assume that the number of pivots choosing for worst case is 50 for X dimension and 30 for Y dimension. We also define the number of matching between the two kinds of regions is 2*(13*13). The LPM algorithm reduces the half of matching process by defining the exact matching area. It also assures the lower number of pivots choosing for partition the routing space. [2]

In the second layer, the specific decision of the region's selection performs to calculate the matching data between the three sets. This layer also supports the subtracted region lists. These lists subtract from the input regions set for next matching calculation. The classification of regions carries out in the region classifier algorithm. The actual matching between the updater regions and the subscriber region execute in intersection calculation algorithm. The subtracted region lists use to reduce the next calculation. [3]

The Layer partition-based matching algorithm consider in 2-dimensional routing space as three different ways. The first method uses the same number of input regions in each dimension. The final overlapping information can get by using AND operation between the overlapping result of two dimensions. [4]

The number of input regions in each dimension of second method is different. The input regions in Y dimension depends on the overlapping result of the X dimension. If some of input regions in X dimension are not overlapped, they cannot include in the input regions of Y dimension. The final overlapping information can produce the result of overlapping matrix in Y dimension. [4]

The third method is piggyback the result of matching result of the X dimension. Before the making decision for Y dimension, the LPM algorithm needs to check the matching result of X dimension. The final overlapping information decides by Y dimension without combing the overlapping results of two dimensions. Three methods of the LPM algorithm will complete when all dimensions are covered. [4]

5. Performance Analysis

For the matching algorithms of DDM, the impact of network speed on the algorithm does not care and actually, there are no messages transferred in the network in all of the approaches. Thus, a single computer used to make experiments. As the performance of the DDM execution time for the matching process is measured with Microsoft Windows 8 with 2.90GHz Intel(R) Core (TM) i7 CPU and 8GB memory. One of the important experimental parameters is the number of regions. The overlap rate defines as the proportion of the scene volume occupied by the regions. Therefore, we define the overlap rate as shown in equation 1:

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

where $\sum$ area of regions = number of regions * high of region * width of region. If the routing space is 100*100 and one region is 1 * 1, where the number of regions is fixed at 100, the overlap rate is

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

5.1 Theoretical Analysis on Computational Complexity

To analyze the computational complexity of the LPM algorithm, we suppose that there are N regions with the number of dimensions, d=2 in the multidimensional space. The optimal pivot algorithm requires O(N) computation for the size of region is M. We assume that the size of region and the number of dimensions is constant. The first layer partition algorithm requires O(N) computation. The total number of recursions for matching algorithm requires O (log N) computation.

Moreover, the second layer partition algorithm also needs O(n) computation and the matching process of comparing the intersection of regions between partitions requires O (n^2) computation (where n is the number of regions in each partition). The complexity of the intersection calculation procedure is proportional to n. It seems that the most important points are the exact matching partitions. It is obvious that the number of regions, n, is a determinant factor. Because the overlap information of all regions obtain by the pivot partition, it is not necessary to compare their overlap information in the left and right partitions of pivot partition. Therefore, the computational complexity of the LPM algorithm is n^2 x N x O (log N) computation. If the number of regions, n, is normally very small in a large-scale spatial environment, so the LPM algorithm should be very efficient. Therefore, the actual computational complexity depends on how the exact matching partition well achieved.

5.2 Performance of DDM Algorithms

The regions distribute randomly across the routing space 10000 * 10000. The number of regions is differing from 1000 regions to 15000 regions.

5.2.1 Performance Analysis of LPM Using Same Size Region. The performance analysis based on the overlap rate 0.01, 0.1 and 1. The figure 6 and 7 shows the execution time for the matching process in four other algorithms and LPM when the overlap rate is 0.01. The figure 8 and 9 shows the execution time for 0.1. For the overlap rate 1 is shown in figure 10 and 11.

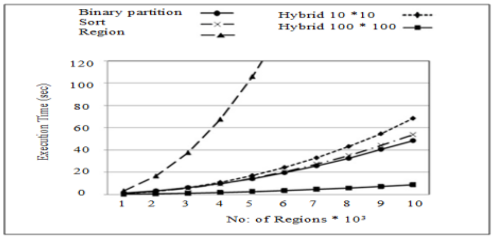

Figure 6: Performance of Overlapping Rate 0.01

It seems that the hybrid approach with 100 * 100 grid cells always has the best performance. The binary partition-based matching algorithm outperforms the other matching algorithms when the overlap rate is 0.01.

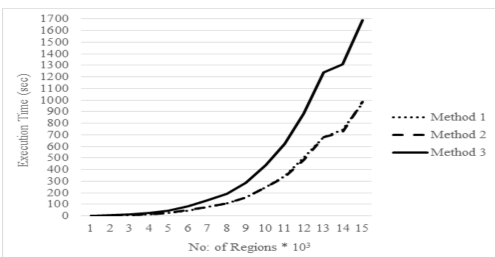

Figure 7: Performance of Overlapping Rate 0.01 of LPM

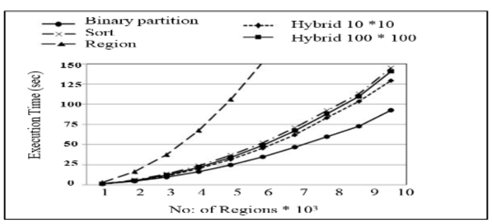

Figure 8: Performance of Overlapping Rate 0.1

In figure 8, the computational overhead of hybrid approach degrades significantly when the number of regions is higher.

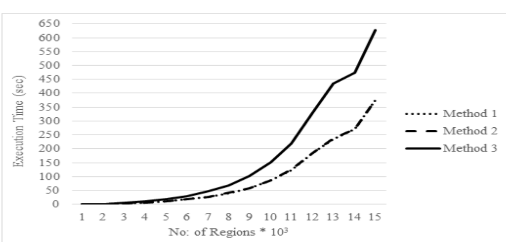

Figure 9: Performance of Overlapping Rate 0.1 of LPM

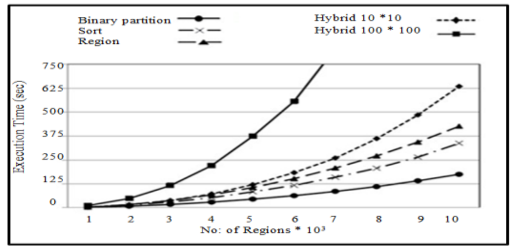

Figure 10: Performance of Overlapping Rate 1

With the overlap rate is 1, the binary partition-based algorithm performs well. On the other hand, the sort-based algorithm performs better, except the binary partition-based algorithm in the overlap rate is 1. When the number of regions increases and the overlap rate is high, the performance of the region-based algorithm becomes increasingly better than the other overlap rate. From all of the figures, we know that the hybrid approach with 100 * 100 grid cells has an extremely big computational overhead for the matching process.

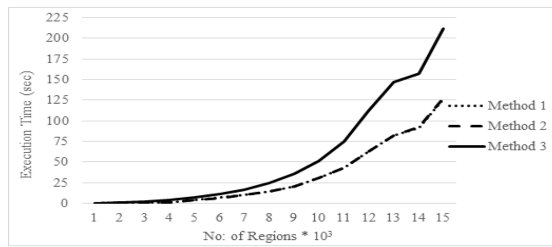

Figure 11: Performance of Overlapping Rate 1 of LPM

For the Figure 7, 9 and 11, the performance of the LPM algorithm analyze on same size regions, which generate randomly. The three methods are of LPM is not the best choice when the overlapping degree is relatively low, but it has the advantage of the matching time when the overlap rate is high. The performance of the first method and the second method are nearly the same. The input region list for second dimension cannot affect the overall matching process. The best method of LPM is the first method. According to the analysis results, it is proved that the execution time of same size regions can be reduced about two third than the previous matching algorithms for the overlapping degree 1.

5.2.2 Performance Analysis of LPM Using Different Size Regions. To define the size of each region base on overlap rate 0.01, 0.1 and 1. To generate the different size region upon the routing space, the equation 2 is used.

$$\text{region size} = \sqrt{\frac{\text{area of space} * \text{overlap rate}}{\text{number of regions}}} \quad (2)$$

The number of regions is the fifteen different sizes from 1000 to 15000. All methods of LPM algorithm are more efficient than the existing matching algorithms of DDM at any overlapping degree using different size regions. The main advantage of this algorithm is its support for scalability very well, when the overlapping degree is large.

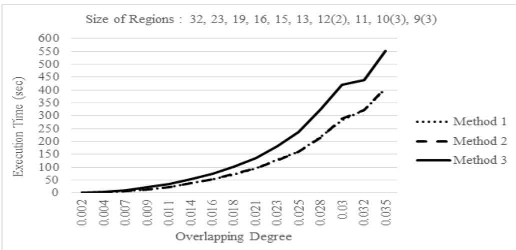

Figure 12: Different Size of Regions Generated by Overlapping Rate 0.01

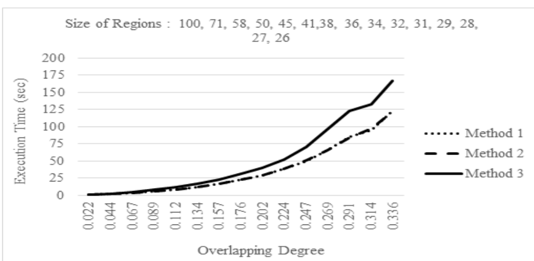

Figure 13: Different Size of Regions Generated by Overlapping Rate 0.1

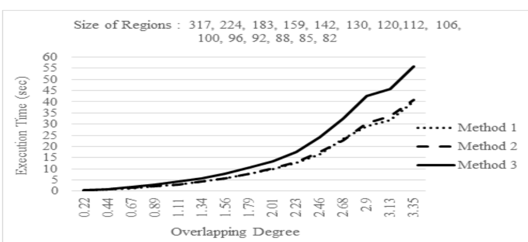

Figure 14: Different Size of Regions Generated by Overlapping Rate 1

6. Conclusion

The layer partition-based matching algorithm is very useful and efficient for the different size of the regions. The LPM algorithm can improve the efficiency and performance by the right choice of the optimal pivot algorithm. The matching process can decrease the number of comparing regions between the pivot partition and left partition and pivot partition and right partition. The number of regions in the projected region list can reduce over and over again by using subtract list. The LPM algorithm does not need the partitioning to cover all regions. According to the analysis result, the LPM algorithm of method one is better than the previous algorithms when the overlapping degree is higher. It supports the best matching result with different region sizes. The final overlapping result can obtain in a timely and efficient manner.

7. References

[1] M. T. Nwe Nwe and T. Nay Min, "Optimization of Region Distribution Using Binary Partition-based Matching Algorithm for Data Distribution Management", International Journal of Engineering Research and Technology (IJERT), Vol. 2 Issue 2, February 2013, pp. 1582-1587.

[2] M. T. Nwe Nwe and T. Nay Min, "Dynamic Pivot for Layer Partition-based Matching Algorithm of DDM based on Regions Distribution", Proceedings of the 4th International Conference on Science and Engineering (ICSE), Yangon, Myanmar, December 9-10, 2013.

[3] M. T. Nwe Nwe and T. Nay Min, "Layer Partition-based Matching Algorithm of DDM", Proceedings of the 3rd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI), Singapore, February 11-12, 2014.

[4] M. T. Nwe Nwe and T. Nay Min, "Layer Partition-based Matching Algorithm of DDM based on Dimension", International Journal of Advanced Computer Technology (COMPUSOFT), Vol. 3 Issue 4, April 2014, pp. 691- 695.

[5] R. Come, T. Gary and S. C. Tay, "A Sort-Based DDM Matching Algorithm for HLA", ACM Transactions on Modeling and Computer Simulation, Vol. 15, Issue 1, January 2005, pp. 14-38.

[6] T. Gary, Z. Yusong and A. Rassul, "A Hybrid Approach to Data Distribution Management", Proceedings of the 4th IEEE International Workshop on Distributed Simulation and Real-Time Applications, San Francisco, CA, August 17-25, 2000, pp. 55–61.

[7] A. Junghyun, S. Changho and G. K. Tag, "A Binary Partition-based Matching Algorithm for Data Distribution Management", Proceedings of Winter Simulation Conference (WSC), Phoenix, AZ, December 11-14, 2011, pp. 2723-2734.

[8] A. Rassul, M. Farshad and T. Gary, "Optimizing Cell-size in Grid-based DDM", Proceedings of the 14th Workshop on Parallel and Distributed Simulation, Bologna, Italy, May 2000, pp. 93–100.

[9] G. Tan, R. Ayani, Y. Zhang and F. Moradi, "Grid-based data management in distributed simulation", Proceedings of the 33rd Annual Simulation Symposium, Washington, DC, April 2000, pp.7–13.

[10] J. Yu, R. Come and T. Gary, "Evaluation of a Sort-based Matching Algorithm for DDM", Proceedings of the 16th Workshop on Parallel and Distributed Simulation, Washington, DC, May 2002, pp. 68–75.