Implementing Data Distribution Management System Using Layer Partition-based Matching Algorithm

Nwe Nwe Myint Thein [1 +]

[1] Lecturer, University of Information Technology, Yangon, Myanmar

Abstract. Simulation has become a popular tool to study a broad range of systems. The growing number and quality of simulation software requires expertise for their evaluation. In High Level Architecture paradigm, the Runtime Infrastructure (RTI) provides a set of services. Data Distribution Management (DDM) service reduces message traffic over the network. DDM services are used to reduce the transmission and receiving of irrelevant data and aimed at reducing the communication over the network. Currently, there are several main DDM filtering algorithms. The paper describes practically testing result for the data distribution management (DDM) service of the Battleground simulation with the dynamic fighters by using layer partition-based algorithm. The layer partition-based matching algorithm is based on divide and conquers approach. It selects the dynamic pivot by detecting the regions distribution on the routing space. This system intends to detect the movement of the fighter objects, searches overlap between the fighter object and every battalions (extents). It is large-scale distributed simulation in order to minimize subsequent computations and algorithm complexity. The developed system can be used not only in the research purpose but also in the real-world distributed applications. It provides the low computational time and exact matching result.

Keywords: Modelling and Simulation, Data Distribution Management, Matching Algorithm, Layer partition-based Matching Algorithm

1. Introduction

HLA (High Level Architecture) is a software architecture which can be reusable for execution of distributed simulation applications. HLA consists of rules, interface specification and object model template. The interactions between federation and federates are governed by an interface specification with the Runtime Infrastructure. Object Model Template (OMT) is a role to document major information about simulations. HLA presents a framework for modelling and simulation within the Department of Defence (DoD). The goal of this architecture is to interoperate multiple simulations and facilitate the reuse of simulation components. HLA allows interconnection of simulations, devices, and human operators in a common federation. It builds on composability, letting designer construct simulations from pre-built components. Each computer-based simulation system is called a federate and the group of interoperating systems is called a federation. HLA specifications—incorporated as IEEE 1516 standard—were developed to provide reusability and interoperability. [1]

Modelling and simulations(M&S) have always been a major part of human history. Modelling can be defined by "the process of producing a model; a model is a representation of the construction and working of some system of interest" and simulation can be defined as "A simulation of a system is the operation of a model of the system" simply. M&S is used to simulate real system's objectives by modelling components, simulation steps and process and implementing produced models in a time flow. The area of M&S was extended from War Game to Task Request, Weapon Acquisition, Decision, Analysis and Military Training. Also, efficient operation of massive simulation and interoperability between complex systems has been

studied. [2] The remainder of the paper organizes as follows. Section 2 describes the theory background of DDM and matching algorithms. The layer partition-based matching algorithm(LPM) explains in section 3. Section 4 represents the simulation model by using LPM algorithm. Finally, section 5 offers conclusion.

2. Literature Review

2.1. Data Distribution Management

Data Distribution Management (DDM) is a set of services defined in HLA to distribute information in distributed simulation environments. HLA's 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 supports by the function of RTI, which is similar to a distributed operating system. The evolution of the DDM service 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. Therefore, many DDM filtering algorithms are proposed. [2,3]

2.2. 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. If there is N update regions and M subscription regions. "There are N*M pairs t check in the worst case". [3]

2.3. 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. Although the overlapping information is not exact, the grid-based algorithm can reduce the computation complexity than the region-based algorithm. 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. [4,5]

2.4. 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. 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. [6]

2.5. Sort-based Algorithm

The sort-based algorithm used a sorting algorithm to compute the intersection between update and subscription regions. The algorithm's performance degraded when the regions are highly overlapped, and it needed to optimize the sorting data structure for the efficient matching operation. [7,8]

2.6. Binary Partition-based Algorithm

The binary partition-based algorithm takes a divide-and-conquer approach. It performs binary partitioning which divides the regions into two partitions that entirely cover those regions according to the midpoint (pivot value) of the routing space. It matches and compares the pivot partition with the left and right partitions. The overall overlap information can be obtained by combining the overlap information of each dimension. The selection of the pivot partition is a determinant factor of the performance of the algorithm. The overlap information of all region is obtained by the pivot partition. If there is no overlapping at all in these partitions, the comparing process is the most time-consuming operation. With the greater overlap rate, the binary partition-based algorithm performs well. [8]

3. Layer Partition-based Algorithm

Layer partition-based matching algorithm (LPM) is executed in dimension by dimension. This algorithm accepts all regions of the routing space. It works layer by layer for the regions. It contains many sub algorithms that perform the specific function. It firstly chooses the most suitable pivot. Then it selects the region generally in the first layer. In the second layer, the specific decision of the region's selection is performed. This layer also supports the subtracted region lists. 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 in the intersection calculation algorithm. The number of input regions in each dimension is the same. The final overlapping information can be produced by observing the result of two matrixes. The LPM algorithm is completed when all dimensions are covered. [9]

The performance of LPM algorithm analyses on same size regions and different size regions, which generate randomly. The execution time of same size regions can be reduced about two third than the other matching algorithms for the overlapping rate is 1. LPM is more efficient than the existing matching algorithms of DDM at any overlapping degree using different size regions. The main advantage of LPM is its support for scalability very well, when the overlapping rate is large. [10] The overlap rate defines as the proportion of the scene volume occupied by the regions as shown in equation 1:

$$\text{overlap rate} = \Sigma \text{ area of regions} / \text{area of space} \text{ -------------- (1)}$$

where Σ area of regions = number of regions * high of region * width of region.

4. Simulation

The system is implemented for dynamic fighter simulation object moving in a battleground. There are two kinds of entities: fighter objects and federates (battalions). The system checks the movement of fighter at each time step and calculate the overlapping algorithm. When overlapping is detected, it informs the corresponding battalion. It is implemented the simulation using java programming language. The system consists of four distributed federates and communicates with them via RTI.
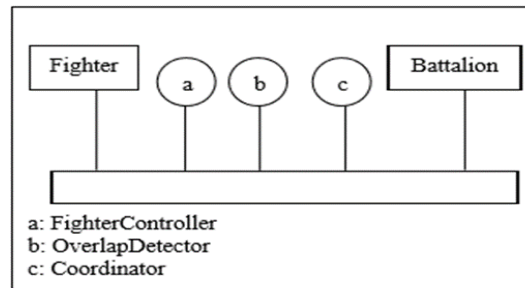

Fig.1: The System Architecture

```
Start Simulation
Initialization
Initialize the number of the fighter objects, check valid number of fighter for each boundary and
generate random number (x,y coordinate) for the fighters
End Initialization
for t=t0+dt, t0+2dt, ......, do
    begin
        • FighterController cpatures the movement of fighters and sends it to RTI
        • OverlapDetector calculate the overlapping between fighter objects and battalions
        • RTI DDM determines subscription region and inform the fighter information
        • Coordinator connects between the actual overlapping subscriber (battalion) and the
            head office server and send fighter
        • Each subscriber received data about the overlapped fighter
    end
endfor
```
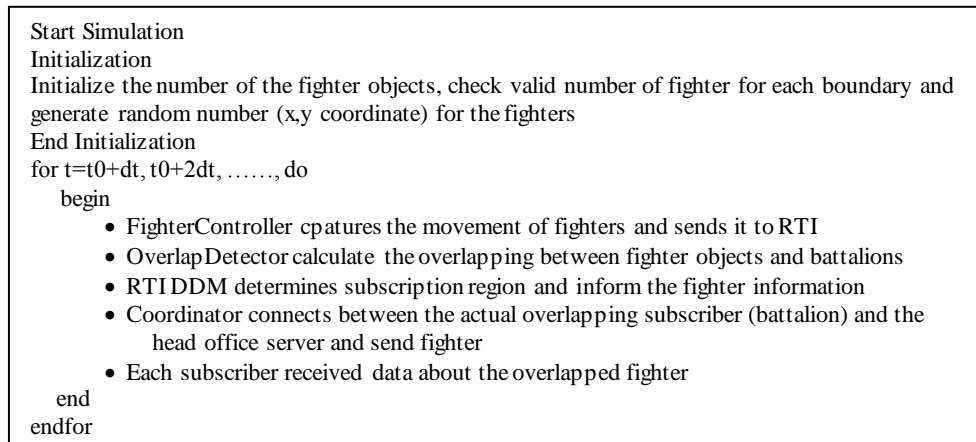
Fig. 2: A time-stepped Distributed Simulation Algorithm

In order to perform the overlapping detection, the system comprises three sub-models: FighterController, OverlapDetector and Coordinator shown in Fig 1. FighterController is used to control the moving of the user defined fighter objects. OverlapDetector uses layer partition-based matching algorithm to detect the

overlapping condition between fighter objects and 12 battalions. Coordinator performs the message transportation duty. The Fig 2 shows the algorithm incorporated in the system for performing the time-stepped simulation of the battleground. In the simulation, twelve battalions are grouped in order to perform the relevant actions. Assume that these groups are in distributed manner shown in Fig3.
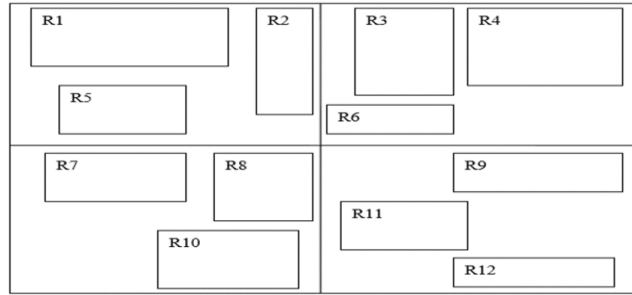

Fig.3: Twelve Regiments Divided into Four Co-operated Group

In this platform, the fighter objects are simulated using the FighterController sub-model. This sub-model is responsible for the creation and deletion of the simulated fighters from the runtime infrastructure. It is assumed that the fighter moves constant velocity and constant direction for several time-steps. The position and movement of the fighter can be detected by investigation of FighterController. The regional battalions act as subscribers and fighter object acts as updater and these processes are described in Fig 4.
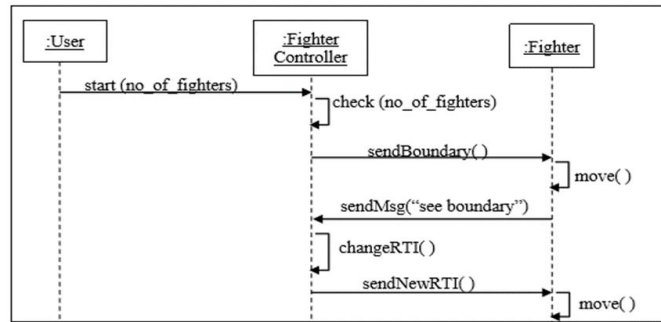

Fig.4: Sequence Diagram for the Action of FighterController Component

In Fig 5, the data filtering strategy of OverlapDetector is presented. The position of the fighter objects may change at every time steps. It searches the overlapping information between the battalions and the fighters by using LPM. When it knows the overlapping information, it connects Local DDM_Manager of the regional battalions (federates). Then it sends the command directly to overlapping federates. If overlapping happens, it produces actual overlap battalion name and sends information to that battalion. Therefore, the battalion may perform required action in timely manner.
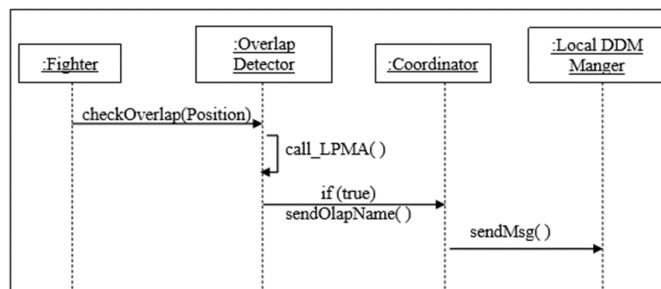

Fig.5: Sequence Diagram for the Action of OverlapDetector Component

The coordinator sub-model is responsible for communication between battalions and their local DDM managers. In the proposed system, it is to be assumed that four federate groups are located in distributed manner. Each group has three battalions. In order to connect them Coordinator uses RMI methodology.

5. Conclusion

The broadcasting mechanism employed in Distributed Interactive Simulation (DIS) standards generates unnecessary network traffic and is unsuitable for large scale and dynamic simulations. Efficient data

distribution is an important issue in large scale distributed simulations with several thousands of entities. In this paper present the simulation model of layer partition-based matching The proposed algorithm also provides the minimum matching cost between the updaters and the subscribers within the routing space. The system can be used not only in the research purpose, but also in the military and civil applications.

6. Acknowledgements

7. References

[1] G. J. Jense, N. H. L. Juijpers and A. C. M. Dumay, "DIS and HLA: Connecting People, Simulations and Simulators in the Military, Space and Civil Domains", Proceedings of 48th International Astronautical Congress, Italy, 1997

[2] B. Azzedine and L. Kaiyuan, "A Novel Approach to Real-Time RTI based Distributed Simulation System", Proceedings of the 38th Annual Symposium on Simulation, April 4-6, 2005, pp. 267-274

[3] L. M. Katherine and S. S. Jeffrey, "Data Distribution Management in the HLA Multidimensional Regions and Physically Correct Filtering", Proceedings of Spring Simulation Interoperability Workshop, No. 97S-SIW-053, March 1997

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

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

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

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

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

[9] M. T. Nwe Nwe and T. Nay Min, "Layer Partition-based Matching Algorithm of DDM", Presented at the 3rd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI), Singapore, February 11-12, 2014, pp. 34-39.

[10] M. T. Nwe Nwe, "Performance Analysis of Layer Partition-based Matching Algorithm in Data Distribution Management", Proceedings of the 2nd International Conference on Advanced Information Technologies (ICAIT), Myanmar, November 1-2, 2018, pp. 114-119.