

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Study of Complicated Distributed Simulation System Entity's Dynamic Management Technique

¹Cheng-lie Du, ¹Kai Yang and ²Nan Jiang

¹School of Computer College Northwestern Polytechnical University,
Xi'an, 710116, Shannxi, China

²Kunming Shipborne Equipment Research and Test Center
Kunming, 650051, Yunnan, China

Abstract: Military-oriented simulation technology has developed toward complicate and distribute's direction recently. Complex means a wide range of subsystem and has the role of multi-level interaction. For this kind of system, dynamic management and entity scheduling between layers and in layers is more complicated. It introduces the concept of node's accommodation degree, designs entity scheduling module and migration module based on simulation supervisory control mechanism and ensures the normal operation of simulation system. At last, it verifies this article under the condition of interconnection between HLA/ RTI and real-time network. This research method provides a good reference for other entity dynamic management of heterogeneous network simulation.

Key words: Federation, reflective memory network, wrapper, real-time simulation

INTRODUCTION

The system of modeling and simulation can be divided into modeling technology, modeling and simulation support system technology and simulation application technology (Li and Chaen, 2002). The establishment of support platform has significant meaning for saving resources, improving efficiency and saving cost. Recently, simulation technology faced to military-oriented filed is developing to the direction of complex and distribution. Complex means a wide range of subsystem and has the role of multi-level interaction. This kind of system often constructs based on heterogeneous network environment. It includes heterogeneous network, heterogeneous support environment software and heterogeneous operating system. For this kind of system, Hierarchical heterogeneous resolves the request of different data's interactive performance but it makes the communication between different layers become abnormally complicated and analysis focused on communication efficiency of different layers' entity interaction becomes difficult. These features decide interlayer and in-layer's entity scheduling and dynamic management be more complicated.

Accordingly, this study introduces the concept of node's accommodation degree and design entity scheduling module and transfer module based on

simulation supervisory control mechanism and ensures the normal operation of simulation system. At last, it verifies this article under the condition of interconnection between HLA/ RTI and real-time network.

BRIEF INTRODUCTION OF COMPLICATED SIMULATION NETWORK ENVIRONMENT

RTI: In October, 1995, U.S. Department of Defense in their goal of modeling and simulating master plan provides for making a common technical framework for the modeling and simulation in the field of defense. HLA (Huang *et al.*, 2003) is the core content of framework. HLA is established under the guidance of Defense Modeling and Simulation Office for implementing interactive simulation among many types and the interactive between simulation and command, control, communications, computers and intelligence, facilitating simulation system and simulation components'reuse. High level architecture whose goal is improving the efficiency of establishing module and simulation, promoting system's mutual operation and reusable, reducing the cost of modeling and simulation for complexity large-scale system is not a system realization but is a simulation application system's framework architecture standard (Liao and Liang, 2002). The two key issues which HLA would like to solve are promoting

interoperability among simulation system and useful for different simulation model to reuse in different simulation application.

VMIC: In real-time network application, traditional Ethernet and Ethernet card can not meet the harsh requirement of real-time performance. So real-time optical fiber network based on virtual shared memory occurred. VMIC company' reflection memory real-time network (Jovanovic and Milutinovic, 1999) Rtnet (Reflective Memory) is one of mature product so far. Reflection memory network is a fast real-time network and it can be used to connect every kind of computers. It has many features about strict transmit certainty, predictability, high-speed, light host load, strong adaptable for hardware and software platform, reliable transmission error-correcting ability, supporting interrupt signal transmission and so on. Reflecting memory network is composed by interface board inserted in the two nodes and the fiber connecting interface board (Logan and Theodoropolous, 2001). In every interface board, there is a special storage and they are mapped to the same address space, constitutes the distributed shared-memory. Reflection memory board can be inserted into many bus's main board such as VME, PCI, compact PCI and so on.

COMPLICATED DISTRIBUTED SIMULATION SYSTEM BASED ON RTI REAL-TIME EXPANSION

RTI real-time network expansion: The viewpoint of distributed simulation high level architecture is order to the dependence of most simulators' application implementation, weak mutual operation and reusability among simulators, high cost of development, maintenance and usage, weak verifiable, effectiveness and confidence. HLA is such a kind of framework from architecture and it can cover the referenced every kind of simulation system in M and S field as possible and make full use of mutual operation and reuse. It can also make use of the developing technology to fulfill complicated big system's simulation system. Using HLA's technical system can connect single simulation application to be a large virtual world. From the point of distributed simulation HLA's proposal and application, it focuses on constructing a complex simulation framework and try its best to satisfy various simulation need and its design focuses on completion ability of large-scale simulation system. So we can see that distributed simulation HLA don't consider the strong real-time application in its design stage. From the network support of distributed simulation HLA, it can only support simulation which uses Ethernet as network

foundation but not every kind of real-time network. The network's non-real time make it can not support real-time simulation not well.

Real-time network expansion means connecting real-time network for the need of semi-physical simulation based on reusing RTI framework. Using mature real-time technology to expand original non-real time simulation is the key point of real-time subnet expansion's research.

The analysis of Real-time subnet expansion's application

condition: Semi-physical simulation is a real-time simulation of using physical to replace partial simulation model. Semi-physical simulation is much closer to actual than mathematical simulation. It has good verisimilitude and a good practical value. Real-time property is a basic requirement of semi-physical simulation. With the development of network technology and the appearance of real-time network, semi-physical simulation technology can be used more and more often. Distributed interactive simulation means interconnecting simulation equipment scattered on every part of world by local area network or wide area network using coordinated structure, standard, protocol and database to inform a participated integrated simulation environment. The entity of distributed interactive simulation using network connects different interconnected equipment to construct a integrated simulation environment. Though it can bring semi-physical simulation system into its integrated system, it must produce conflict with the real-time ability of semi-physical simulation as distributed interactive simulation is based on local area network and wide area network. For example, inherent Ethernet and distributed simulation's mechanism can not meet real-time's need despite local simulation model needing large amount of data exchange and high-interaction, which causes delay of whole simulation system's increasing.

Figure 1 gives out topological structure of semi-physical system and RTI's cross-protocol interconnection.

Gateway protocol conversion: Gateway's design is to resolve the problem of distributed simulation real-time expansion, so it must resolve the problem of mutual operation, protocol conversion, data format conversion under heterogeneous network platform. We use such structure to design gateway as Fig. 2.

In Fig. 2, application interface layer charges connecting different system's interfaces in order to implement protocol conversion. Data integration layer implements the function of data format's definition and conversion. Logical integration layer charges control of

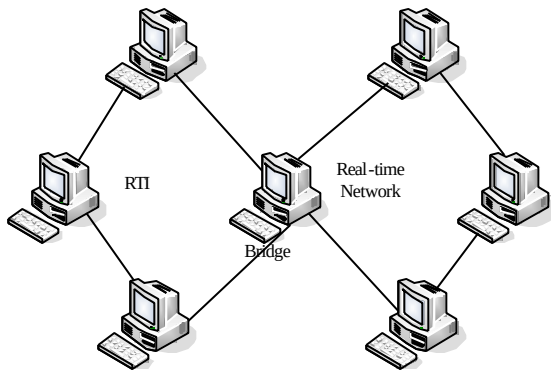


Fig. 1: Bridge between RTI and real-time network

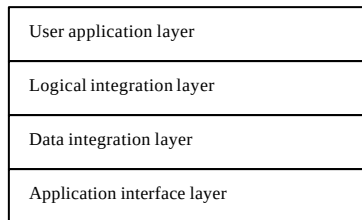


Fig. 2: Diagram of gateway's layers

simulation logic. User application layer is an operation interface that is exposed to users and user configures operation parameter by friendly interface.

TECHNOLOGY OF ENTITY DYNAMIC MANAGEMENT

According to modularization's design idea, entity dynamic management is mainly divided into three modules: Entity scheduling module, monitoring module and entity migration module. The whole framework structure is as follows.

Monitoring module: Monitoring module is an assisted module of entity scheduling system, the main entity of which is collecting service node's state information periodically and calculates every node's node accommodation degree for use of entity scheduling module and entity migration module's decision.

It's responsible for monitor all nodes running in system and all nodes can run on different machines.

It can select monitoring any node running in system.

It displays interface's name, times, maximum time, minimum time and average time in invoking system.

Every node's reading and writing time of network.

Every node's CPU and memory utilization, which can be expressed with curve.

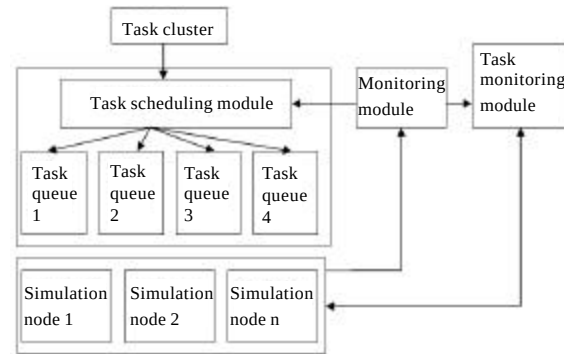


Fig. 3: Entity scheduling system structural diagram

Calculating every node's node accommodation degree.

We can define another variable that is node accommodation degree in order to describe system's load weight. When nodes are added to virtual run-time experimental system, server gives an estimated value which represents simulation node's handling-ability according to the node's soft hardware and other limiting condition, that is node accommodation degree. Node accommodation degree reflects node ability of accepting entity. The definition of node accommodation degree is as follows Formula 1:

$$S'_i = \frac{W_i}{\text{Load}(i)} \quad (1)$$

W_i represents node's performance weight coefficient, $\text{Load}(i)$ represents node's load coefficient, S'_i is proportional to W_i and is inversely proportional to $\text{Load}(i)$. However, the range of node accommodation degree S'_i calculated by above-mentioned method is uncertainty and it is not conducive to the expression and use. So standardizing above-mentioned node accommodation degree and restricting its scope interval between 0 and 1. The concrete method is choosing node accommodation degree's maximum value:

$$S'_{\max} = \max(S'_1, S'_2, \dots, S'_n)$$

$$S_i = \frac{S'_i}{S'_{\max}} \quad (i=1, 2, \dots, n) \quad (1)$$

Entity scheduling device is proportional distribution to every service node according to node accommodation degree in order to implement relative balance which load adapting to node's relative handling ability. For example,

The handling ability of Node 1 is very strong and we get its weight $w_1=2$ by calculating. The handling ability of Node 2 is much weaker and we get its weight by calculating $w_2=1$. When Node 1's entity load rate is 0.8 and Node 2's entity load rate is 0.25, $S_1=2.5$, $S_2=4$, the entity accepting ability of Node 2 is bigger than Node 1. Node accommodation degree provides distribution base for behind entity scheduling algorithm's design.

Entity scheduling module: Entity scheduling module is mainly responsible for entity cluster's reading and periodic entity and schedules entity according to algorithm for the maximum priority of emergency in node in accordance with distributing entity to every node based on distribution algorithm for node accommodation. Entity scheduling module provides entity receiving interface for requesting client firstly. When entity can enter entity scheduling module, scheduling server charges entity cluster's reading and analyzing and maps non-periodic entity to periodic entity. Then it is encapsulated to be entity object and initializes entity's attribute value. Entity scheduling module establishes server node list and gets every server node's load information and node accommodation degree periodically from monitoring module. It chooses the node reference whose node accommodation degree is the biggest to distribute and implements entity's transferring from requesting queue to the some concrete node's entity scheduling queue.

After entity enters every node's scheduling queue, it should calculate entity's dynamic sub-priority level according to entity's relative static attribute and compare all entity's emergency degree in scheduling queue according to the maximum emergency degree's priority algorithm to get the execution of the highest emergency degree scheduling to server client. The flow Figure of node entity scheduling is as Fig. 4.

Entity migrating module: Entity migrating module strategy can be divided into centralized type and distributed type. Centralized type algorithm can be implemented easily and is easy to optimizing the global but is weak in performance of scalability and fault-tolerance while Distributed type algorithm is difficult to implement and it needs every node maintaining latest system status information and is general to optimize the local. But in most of case, Centralized type algorithm has better efficiency than distributed type algorithm. This system adopts the method of combining centralized type with distributed type to implement entity migrating module in order to play these two methods' advantage. From view of the whole virtual experiment system,

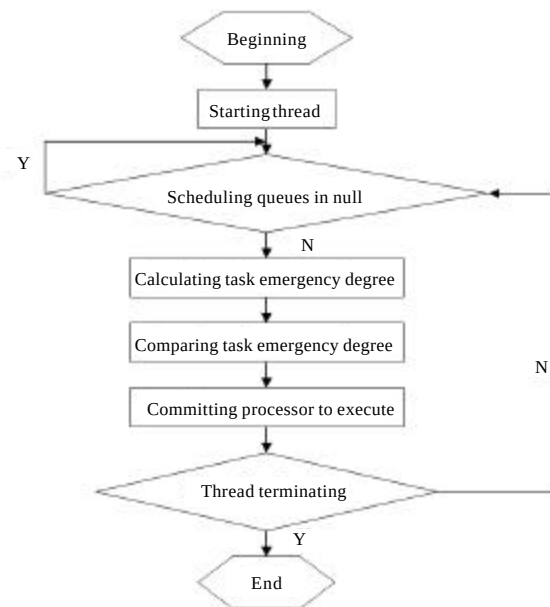


Fig. 4: Flow diagram node entity scheduling

migration belongs to centralized strategy but different sub-network can schedule LAN's migration request independently. Entity migration module is divided into in-network processing sub-module and inter-network processing sub-module.

In-network processing sub-module: Getting in-network node information description form from monitoring module and charging for migration request's processing of same LAN. The concrete process is as follows. When the node becomes overloading-node, it requests entity migration from in-network migration manager. In-network migration module judges the question whether there is light-load machine in network according to node information form. Yes, then in-network. No, then inter-network migration.

Inter-network processing sub-module: Establishing the whole system's node information form and charging for inter-network's migration requesting process. The work process of migration strategic module is as follows. When node starts to entity migration, if in-network migration manager finds there is no light-loading node in network, it requests inter-network migration to inter-network migration manager. If there is light-loading machine in network, it chooses light-loading node as immigration node according to relative network's in-network migration manager and inter-network migration then. If there is no light-loading machine in network, it returns to process

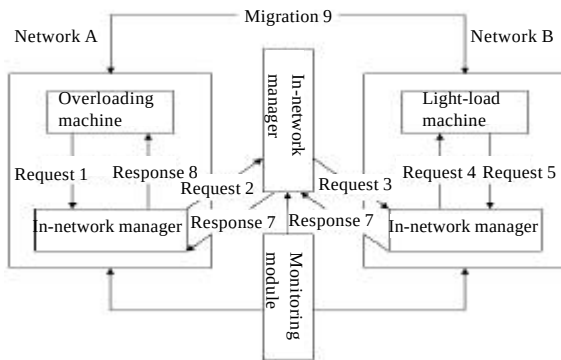


Fig. 5: Schematic Figure of inter-network migration strategy

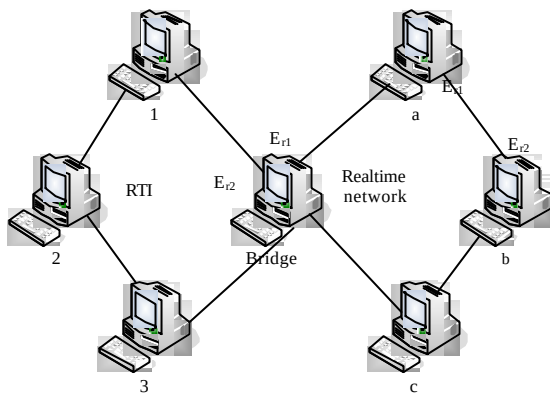


Fig. 6: Entity network topology of migration

information and entity migration is fail. The inter-network migration can improve different LAN's simulator's status that load is unbalance. Compared to in-network migration, migration requesting from different LAN may be interaction. So we should take proper coordination strategy. Figure 5 is inter-network migration strategic Figure.

EXAMPLE VERIFICATION

For vivifying entity migration strategy's efficiency, we run a test on internet topology between MAK RTI2.4 and VMIC5565 (Fig. 1). Run entity E_{r1} and E_{r2} on Ethernet's node 1 and node 2. Run entity E_{r1} and E_{r2} on Ethernet's node a and node b. The host which E_{r1} and E_{r2} run on is over-loading. After running is over, we can find some problems according to searching log. That is:

- E_{r1} migrates into bridge node two seconds after beginning of the experiment
- E_{r2} migrates into bridge node ten seconds after beginning of the experiment

Now entity network structure Figure shows as Fig. 6. When entity exceeds host's load, it causes migration and the goal of migration is bridge node which has higher accommodation degree.

CONCLUSION

Though layered heterogeneous solves the need of different data interaction performance of difficult distributed simulation, it makes interlamellar communication become abnormally difficult and the communication efficiency analysis of interlamellar entity communication becomes difficult. These features decides the truth that inter-layer entity scheduling, in-layer entity scheduling and dynamic management are more difficult. Hereby, this study introduces the concept of node accommodation degree and designs entity scheduling module and migration module based on simulation monitoring mechanism. For real-time need of RTI, it designs a entity distribution algorithm based on node accommodation degree. That's distributing the node which has higher priority to the node which has bigger accommodation degree in order to improve entity's real-time. Using algorithm of maximum emergency degree priority solves the problem of single node's entity scheduling. This algorithm combines static priority's scalability with dynamic priority's flexibility. The next step's work is researching simulation system information transmission method's distribution strategy and promoting mixture time strategy under heterogeneous network environment.

REFERENCES

- Huang, X.D., Y. He and B.Q. Jiang, 2003. The research and test of RTI real-time and information throughput. *J. Syst. Simulat.*, 15: 516-518.
- Jovanovic, M. and V. Milutinovic, 1999. An overview of reflective memory systems. *IEEE Concurrency*, 7: 56-64.
- Li, B.H. and Z. Chen, 2002. Modeling and simulation in China-present and future. *Proceedings of the 5th Asian Conference on System Simulation and Scientific Simulation*, International Computing, November 3-6, 2002, Shanghai.
- Liao, Y. and J.H. Liang, 2002. *Real-Time Simulation Theory and Supporting Technology*. National Defense Science and Technology University Press, China.
- Logan, B. and G. Theodoropolous, 2001. The distributed simulation of multiagent systems. *Proceedings of the IEEE*, Volume 89, February 2001, IEEE., pp: 174-185.