Advances in Intelligent Systems Research, volume 153

International Conference on Applied Mathematics, Modeling and Simulation (AMMS 2017)

An New Method of Distributed Communication Framework Based on Simulation

Jianning Lin, Chenyan Kong, and Peng Cui

Science and Technology on Information Systems Engineering Laboratory, Nanjing 210007, China

***Abstract*—According to the requirement of communication link simulation in simulation tests, a distributed simulation communication framework was designed. The framework can dynamically load different communication efficiency computation models according to the requirements of tests, and the communication efficiency computation models exist in DLL format. Then, the deployment, initiation and operation of the framework are explained in this paper; communication efficiency computation models based on the changes of link status were proposed, and the communication efficiency computational cost reduced.**

Keywords-communication efficiency of distributed communication framework

I. Introduction

High-level architecture (HLA) is a general purpose architecture for the simulation design and development for specific simulation tasks. Most of the current distributed simulation systems are constructed based on HLA specifications. Currently, simulation systems usually have distributed structures and contain multiple simulation nodes. In order to enhance the authenticity of simulation, the communication links among different nodes need to be simulated.

Currently, the studies of communication framework mostly focus on the actual network link. Instant communication system based on B/S and C/S structures was reported in 2009 [1] [2], in which Ajax [3] and other technologies were used as client implementation mechanisms primarily serving the information exchange between real systems. Agent-based communication framework was proposed [4] in 2010, in which XML and SOAP technical structures were used for the conversion between different transport protocols in the process of system information exchange. Studies in the simulation field mainly focus on the models of communication efficiency.

II. Component-based simulation communication framework

A. The Structure Of Simulation Communication Framework

Simulation communication frameworks adopt C/S structure, and consist of a server for computation and a client for calling, as shown in Figure 1. The computation server includes the module of communication computation model deployment, the module of communication model initialization, the module of communication simulation computation request processing, and the module for synchronization of the framework’s client end. The client end includes the module for synchronization with the computation server, the module of simulation effect feedback, and the module for node status reporting. In the simulation process, the server of the framework can be deployed on high-performance computers in order to meet the requirement of computation performance.

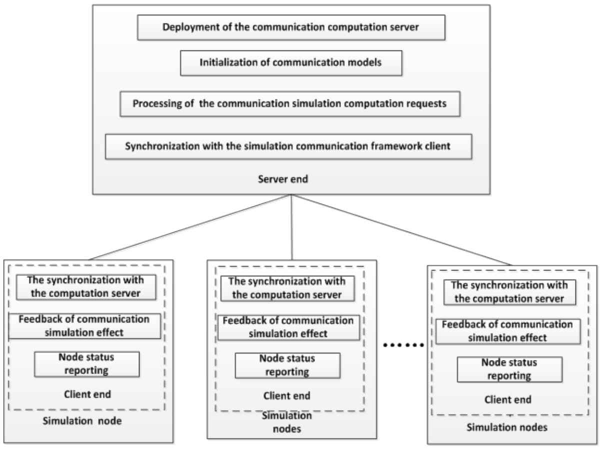
Deployment of the communication computation server
Initialization of communication models
Processing of the communication simulation computation requests
Synchronization with the simulation communication framework client
Server end
The synchronization with the computation server
Feedback of communication simulation effect
Node status reporting
Client end
Simulation node
The synchronization with the computation server
Feedback of communication simulation effect
Node status reporting
Client end
Simulation nodes
......
The synchronization with the computation server
Feedback of communication simulation effect
Node status reporting
Client end
Simulation nodes

FIGURE I. DIAGRAM OF THE SIMULATION COMMUNICATION FRAMEWORK

The simulation communication framework will automatically load and configure the required communication efficiency computation models, and these models should be deployed in the framework in advance. The deployment can be done by developers after the models have been developed. Different communication efficiency computation models can be deployed in a simulation communication framework; they are DLL files and executable programs of processing logic. Each comunication efficiency computation model should have a descriptive file, which is in XML format. This file contains information such as the list of initialization parameters of the model, the storage information of initialization parameters, identification of the model, identification of simulation nodes for this model, and interface parameters of the model.

B. The Structure Of Simulation Communication Framework

After development of the communication efficiency computation models, they are integrated into the communication framework through the module of model deployment, in order to be used for simulation tests. As shown in Figure 2, the model deployment module has the functions of deployment, un-deployment, and re-deployment. The deployment of models refers to the process of integrating the developed communication efficiency computation models into the simulation communication framework, readying them to be called. In addition to deployment, the model deployment module also un-deploy expired models and re-deploy updated

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

models. The communication computation model deployment module is a series of operations on DLL files, including the monitoring interface of local folders, the deployment sub-module, the un-deployment sub-module, and the configuration parameters of model framework. The monitoring interface of local folders is used to receive local or remote DLL files and send these files to the deployment sub-module or the un-deployment sub-module, which deploy or un-deploy these DLL files into the assigned directory on the computation server.

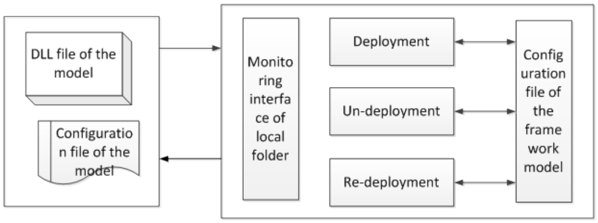
DLL file of the model
Configuration file of the model
Monitoring interface of local folder
Deployment
Un-deployment
Re-deployment
Configuration file of the frame work model

FIGURE II. THE DIAGRAM OF THE FUNCTION OF DEPLOYMENT

In each test with the simulation system, a different communication efficiency parameter computation model will be used. Therefore, after the computation server of the simulation communication framework initiates, the communication efficiency computation model required for the current simulation test will be loaded automatically. Information of the pre-configured load and configuration file is needed when loading, and the information exists in XML format.

Figure 3 illustrates the loading process of communication models on the computation server of the simulation communication framework. As shown in the diagram, the computation server of the simulation communication framework operates by the following procedure.

- Firstly, the computation server of the simulation communication framework is initiated;
- After initiation, the configuration files of the simulation communication models to be loaded will be read;
- If the configuration file exists, it is read to acquire the list of needed simulation communication models and initiation parameters for each model, and these information are then saved in memory; if the configuration file does not exist, the loading process is cancelled and step 6 will be executed;
- The list of loading models is traversed, and DLL files are loaded based on the model initiation path;
- If all the models in the list have been loaded successfully, end this process; otherwise, go to step 4;
- End the loading process.

When the models are loaded, the computation server initiates each loaded model according to the configuration files. The initiation process is as shown in Figure 4.

- The computation server reads the model list, which is saved in the model loading process, traverses the list and read the initiation parameter information of models that have not been initiated; the initiation parameter information include the list of parameters needed for initiation and the default value of each parameter;
- The computation server of the simulation communication framework accesses the backend database and reads the initiation parameters of the model based on the model ID; if the database access fails, go to step 5, end the initiation process and report errors;
- The communication efficiency simulation model is initiated according to the obtained initiation parameters;
- When all the communication efficiency simulation models are initiated, end the initiation process; otherwise, go to step 1;
- End the initiation process.

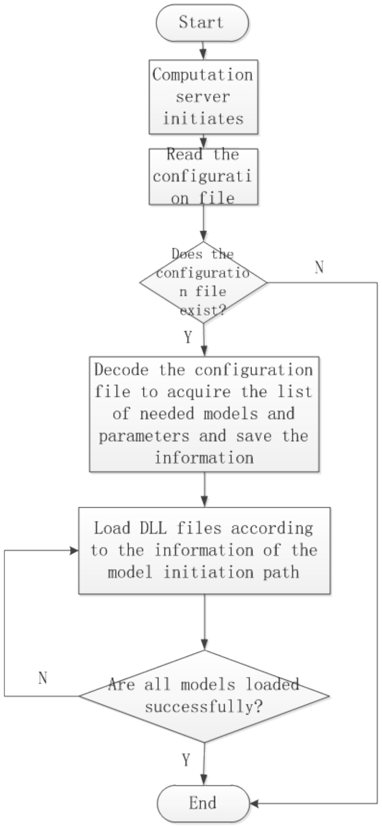
Start
Computation server initiates
Read the configuration file
Does the configuration file exist?
N
Y
Decode the configuration file to acquire the list of needed models and parameters and save the information
Load DLL files according to the information of the model initiation path
N
Are all models loaded successfully?
Y
End

FIGURE III. THE ARCHITECTURE OF NETWORK CENTRIC SIMULATION

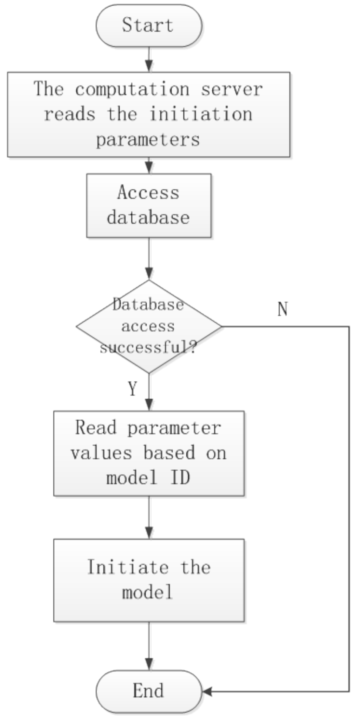
Start
The computation server reads the initiation parameters
Access database
Database access successful?
N
Y
Read parameter values based on model ID
Initiate the model
End

FIGURE IV. THE INITIATION PROCESS OF MODELS ON COMPUTATION SERVER

When the communication efficiency computation models are initiated, the computation server of the simulation communication framework calculates the communication efficiency parameters of each simulated communication link in the test, as shown in figure 7. After the calculation, the computation server of the simulation computation framework synchronizes the results with the clients that are connected with the simulation nodes on the two ends of the link. When the client end receives these parameters, it overlaps the communication efficiency parameters of the messages sent through this link, until the parameters of this link have changed and need to be re-calculated and re-synchronized. This process is shown in Figure 5.

- The computation server loads models;
- The computation server initiates models;
- The computation server traverses the list of connected simulation nodes;
- A simulation node is selected, and the communication link adjacent to this node is traversed; based on the simulation nodes at the ends of this link, the communication efficiency computation model is matched. If this model exists, computation request is sent to this model;
- The computation server receives the parameters of communication efficiency of the communication links calculated by the communication efficiency computation model and synchronizes the parameters with the clients that are connected to the simulation nodes in the link;
- When all the nodes are traversed, end the computation process; otherwise, go to step 4;
- End the computation process.

In the simulation test, the performance parameters of communication links may change. The communication efficiency parameters of a link will change with the change of the performance parameters of this link. Therefore, the communication efficiency parameters need to be recalculated according to the new performance parameters of communication links. The change of the performance parameters can be manually changed via the human-machine interface provided by the sub-module of update of the computation server, or the status information of updated communication links of simulation nodes can be regularly updated to the computation server via the node status reporting module of the client end of the framework.

- When the computation server receives the message of link status changes, it saves this information in the status updating message queue;
- The synchronization module of the client end of the simulation communication framework records the IDs of simulation nodes on the two ends of the communication link;
- The client end traverses all the saved communication efficiency computation models and match the IDs of nodes in step 2 with the simulation nodes of the link in this communication efficiency computation model;
- If the nodes of communication links in this model are the same with the changed nodes, the communication efficiency parameters are calculated according to the changed performance parameters of the communication links; otherwise, step 3 is executed;
- The changed communication efficiency parameters are synchronized to the simulation nodes on the two ends of the communication link via the synchronization module of the client end of the simulation communication framework.
- End the synchronization process

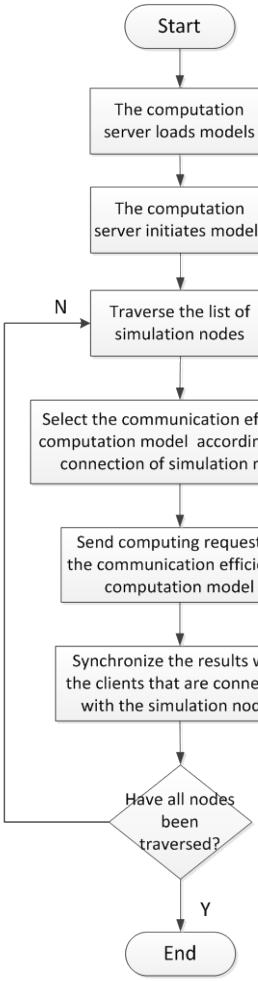
Start
The computation server loads models
The computation server initiates models
N
Traverse the list of simulation nodes
Select the communication efficiency computation model according to the connection of simulation nodes
Send computing request to the communication efficiency computation model
Synchronize the results with the clients that are connected with the simulation nodes
Have all nodes been traversed?
Y
End

FIGURE V. THE COMPUTATION PROCESS OF EFFICIENCY PARAMETERS OF THE COMPUTATION SERVER

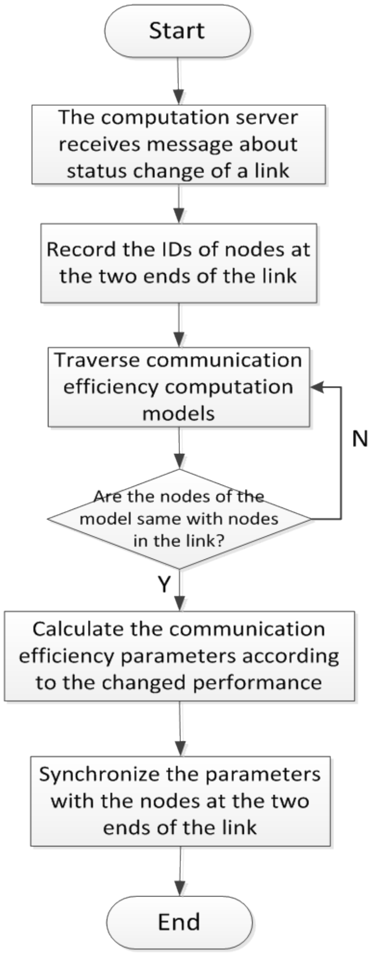
Start
The computation server receives message about status change of a link
Record the IDs of nodes at the two ends of the link
Traverse communication efficiency computation models
N
Are the nodes of the model same with nodes in the link?
Y
Calculate the communication efficiency parameters according to the changed performance
Synchronize the parameters with the nodes at the two ends of the link
End

FIGURE VI. THE PROCESS OF UPDATING THE EFFICIENCY PARAMETERS OF THE COMPUTATION SERVER

III. CONCLUSION

This paper presented a component-based simulation communication framework, which supports the parallel running of communication models with different types of communication links, in order to meet the requirements of simulation systems. Meanwhile, since the parallel running of multiple communication efficiency simulation models will consume massive computing resources and even result in computational delay, a reasonable communication efficiency computation framework was proposed, in order to avoid parallel run of multiple communication efficiency simulation models and reduce the consumption of computing resources.

REFERENCE

[1] The Institute of Electrical and Electronics Engineers: IEEE Standard 1516 for Modeling and Simulation (M&S) High Level Architecture (HLA) ---Framework and Rules, http://www.ieee.org, 2000

[2] Wang, Q. and Ding, Y. W. DSCM:a Service Component Architecture Model Supporting Dynamic Evolution. Journal of Chinese Computer Systems, 2010, 31(1): 130-133

[3] Escoffier, C., Hall R. S., and Lalanda, P. IPOJO:An extensible service oriented component framework[C] // 2007 IEEE International Conference on Services Computng, Salt lake City:IEEE Computer society, 2007:474-481

[4] Li, X. Y. and Yu, Q. A communication framework of multi-agent system. Journal of Chongqing University of Technology, 2010, 24(5): 100-104.Xu L J, Peng X Y. RTI Resource Management System Based on Grid [J]. Journal of System Simulation, Vol 8, No.17, 2005

[5] Xie Y, Teo Y.M., Cai W.T. A Distributed Simulation Backbone for Executing HLA-based Simulation over the Internet [EB/OL]

[6] Zajac K, Bubak M, Malawski M, Sloot P.M. A. Towards a grid management System for HLA-based interactive simulations [A]. In Proceedings of Seventh IEEE International Symposiums on Distributed Simulation and Real-Time Applications[C].23-25 Oct, 2003

[7] Li B H, *et al.* New Distributed Collaborative Simulation System-Simulation Grid [J]. Journal of System Simulation, Vol 20,No 20,Oct,2008

[8] Li Bo-hu, Chai Xu-dong. Networked Modeling & Simulation Platform Based on Concept of Cloud Computing ----Cloud Simulation Platform Journal of System Simulation, Vol.21,No.17,Sep,2009