

2012 AASRI Conference on Modeling, Identification and Control

A Collaborative Simulation Framework based on Distributed Computing Environment

Feng Zhu*, Yiping Yao, Tengfei Hu, Laibin Yan

School of Computer Science, National University of Defense Technology, Changsha and 410073, China

Abstract

Large-scale complex system modeling and simulation involves multidisciplinary knowledge. It is usually necessary to create models using different modeling languages and methods. Therefore, a simulation system built by these models has the characteristics of heterogeneous and hierarchical structure. It will be a great challenge in designing and controlling simulation execution. This paper proposed a hierarchical modeling method and designed a synchronous control algorithm based on synchronous points array. Then we implemented a collaborative simulation framework based on distributed computing environment. The result illustrates this framework provides an efficient mechanism to support hierarchical modeling. With the rate of computing load to communication load increasing, the parallelism will be higher, thus the performance of the simulation system can be promoted.

© 2012 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](#).
Selection and/or peer review under responsibility of American Applied Science Research Institute

Keywords: distributed computing environment, simulation framework, hierarchical modeling method, synchronous control algorithm

1. Introduction

Collaborative simulation is a simulation technology which organizes varied models to work together for analysis and evaluation. The models are usually located in different geographical areas, based on different computer systems, or built with different modeling languages and different modeling tools [1, 2]. The main idea of collaborative simulation is to solve simulation problems within the course of designing complex

* Corresponding author. Tel.: +86-13874868244; fax: +86-0731-84576813.
E-mail address: zhufeng@nudt.edu.cn.

systems [3]. Although some existing simulation tools, to some degree, can solve most analysis and evaluation issues of complex systems in single domain, they lack the support for complex system simulation in multidomain [4]. With the increase of sophistication and complexity within complex systems, it is difficult to solve the problem only use one kind of simulation tools [5]. Therefore, collaborative simulation technology for complex system simulation in multidomain becomes an important trend [6, 7].

Recently, a collaborative simulation system often combines hydromechanics, aerodynamics, mechanical control models to achieve united simulation. It involves various computing models with multidisciplinary knowledge [8]. Besides, each computing model is developed using various modeling languages and simulation tools. As a result, the simulation system composed of those models has the characteristics of heterogeneous and hierarchical [9, 10]. It brings a great challenge in the design and corporate control of simulation systems. To solve this problem, the paper proposed a collaborative simulation framework based on distributed computing environment. The results illustrate this framework provides an efficient mechanism to support hierarchical modeling. Moreover in the distributed computing environment, the performance of the collaborative simulation system can be improved.

The remainder of this paper is structured as follows: in section 2 we explain our collaborative simulation framework in detail. In section 3 we give the analysis with experiment. Finally, our conclusion will be made with an indication of the future work.

2. Collaborative simulation framework

This paper has first studied the key technologies of collaborative simulation, then designed and implemented a collaborative simulation framework which is based on distributed computing environment. Computer nodes are classified as a simulation control node and several computing nodes in our framework. Each computing node contains at least one computing model. Computing models run on computing nodes and pass output parameters, status information to the simulation control node though TCP/IP protocol. In the running, the simulation control node receives output parameters and model status from the previous computing model, then passes input parameters to next computing model and schedules it to run. The collaborative simulation framework shows as figure 1.

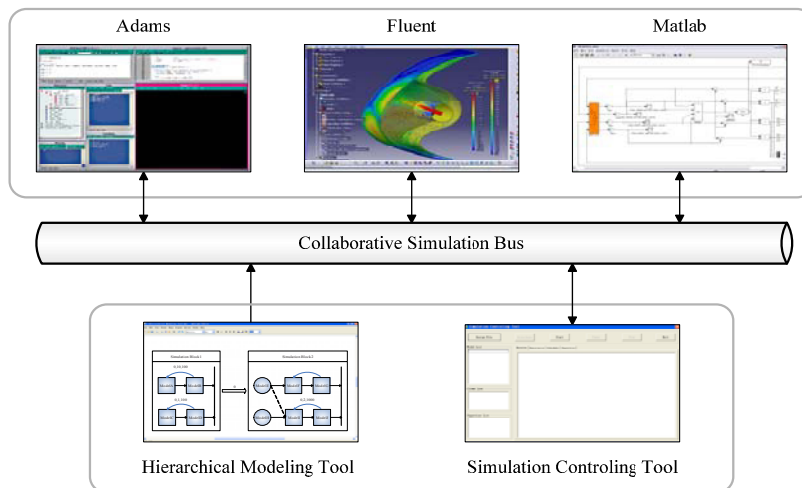


Fig. 1. The collaborative simulation framework

2.1. Hierarchical modeling approach

A collaborative simulation often combines hydromechanics, aerodynamics and mechanical control models to achieve united simulation. It involves multidisciplinary computing models, such as *CAE* model, engineering algorithm model and unified physical model. In order to implement the distributed developing, debugging and packaging and ensure those models fine-grained and high quality of reliability. Those models are built by experts from different research domains using different modeling languages and methods. For instance, *CAE* models are usually developed using *Fluent*, *CFX*, *Ls_Dyna*, *Nastran*, *Abaqus*. Engineering algorithm models are usually developed using *Matlab*, *Fortran*, *C* and so on. Unified physical models are usually developed using *Modelica*. Those models can be assembled hierarchically to construct a complex simulation system model.

Establishing a precise system model is a hard and time-consuming work. To achieve the flexibility and generality, and to support building varied and hierarchical system model, we use graphic elements to represent computing models. The input-output relationship is described by the lines connecting between these elements, as shown in figure 2. According to the order of execution, we group the models into different simulation blocks. Each simulation block contains several loop branches whose starting time and end time are the same. Each loop branch contains multiple computing models. Because of the differences of executive mode, we divide the model into once model or iterative model. Once model is the model that exits immediately after it finishes one step, represented by circle. Iterative model is the model that without exiting until to the end of this loop branch, represented by pane. We use one-dimensional unidirectional solid arrow to represent the scheduling relationship between computing models. One-dimensional unidirectional dotted arrow only indicates the input-output relationship while two-dimensional unidirectional solid arrow represent the order of execution between different simulation blocks. Each loop branch is denoted by the starting time, time step and the end time. Vertical thick lines mark the synchronization time. Because of the execution order between computing models, and the difference of time steps, it is necessary to synchronize among loop branches, to ensure the validity of model execution and parameter transmission. The establishment of the entire simulation system model uses mouse dragging. It is so convenient to improve the flexibility of building the complex system simulation model.

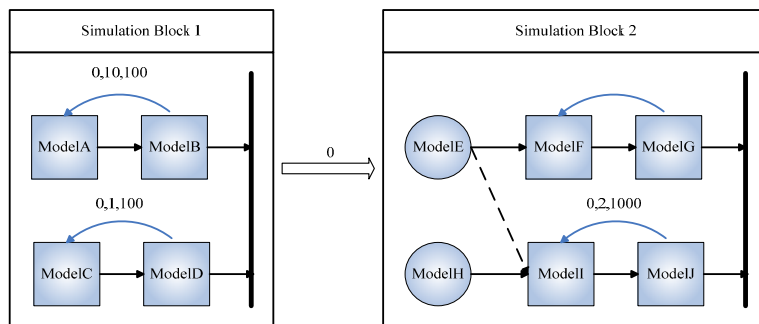


Fig. 2. The hierarchical modeling approach

2.2. Synchronous control algorithm

The collaborative simulation synchronous control algorithm assembles multithreads, a synchronization points array and a waiting queue to control the execution of the simulation system model. A thread for each

loop branch will be created to schedule computing models to run. In order to reduce the times of synchronization, only when it runs to a synchronous model (i.e. the last model of a loop branch, such as *ModelB*, *ModelD*, *ModelG* and *ModelJ* shown in figure 2), the simulation needs to be synchronized.

First, we create a synchronous points array, where the array elements are the multiples of the second small-time step of all the loop branches in a same simulation block. (For example, suppose that a simulation block includes six loop branches, and the time steps of each loop branches are 3, 1, 3, 6, 12 and 1. The end time of all the loop branches in this simulation block is set to 120. Thus the elements of the synchronous points array are (3, 6, 9, 12...120)). Each element of the synchronous points array is a synchronization time. All the loop branches need to be synchronized at each synchronization time.

Next, we set up a waiting queue. The element of this queue is the number of a loop branch whose current simulation time plus the time step exceeds the current synchronization time. We use the waiting queue to control the execution of each loop branch. When the simulation time advanced, if the simulation time of a loop branch in the waiting queue plus the time step less than the current synchronization time, the loop branch will be resume. The collaborative simulation synchronous control algorithm shows as figure 3.

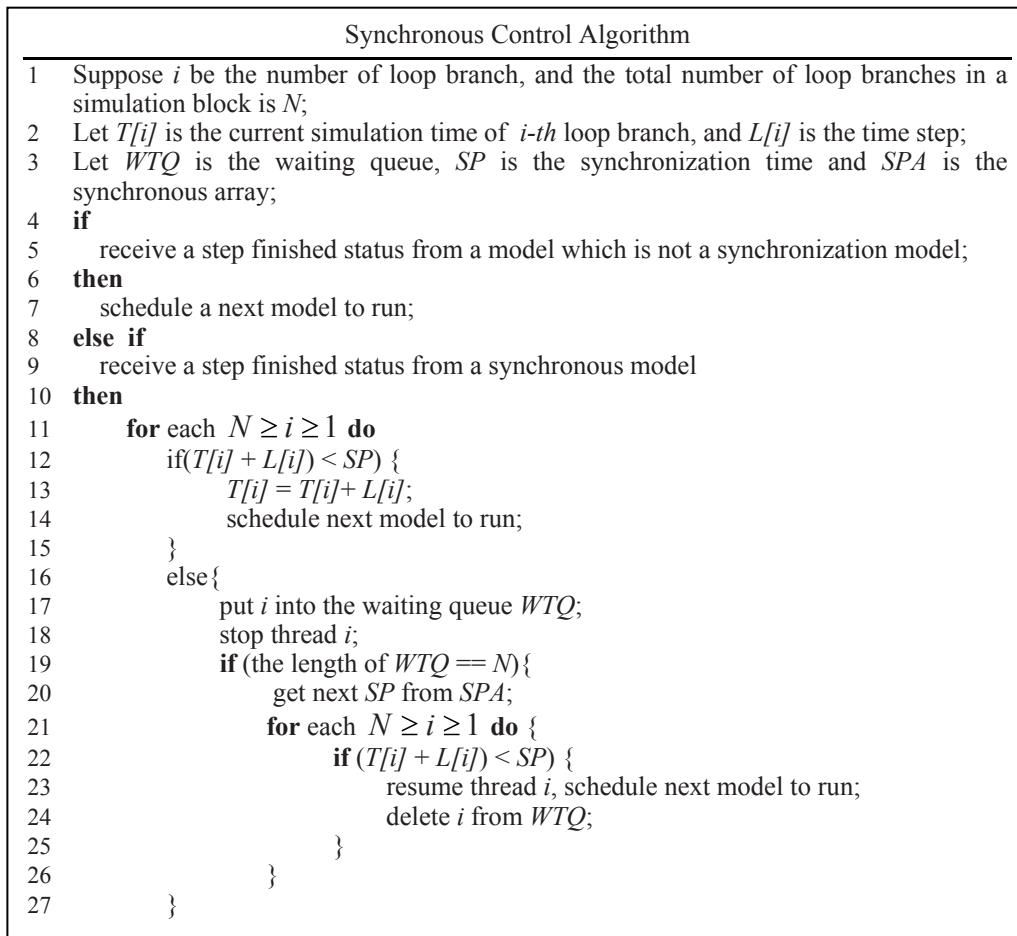


Fig. 3. The synchronous control algorithm

3. Experiment Analysis

In order to analyze the performance of our simulation framework, we distribute the computing models which compose the system model to different computer nodes in distributed computing environment. There are 15 parameters of average that passed among computing models. The data type is double float. Each loop branch iterative 100 times. The collaborative simulation controlling module runs on one computer node, and the models run on other different computer nodes. The system model is composed of 8 loop branches which includes 16 computing models. The Y-axes means the execution time of one model iterating once. There are 3 circumstances in our experiment: (A) no computing load; (B) the rate of computing to communication is 3:1; (C) the rate of computing to communication is 30:1. According to this, we research the impact of the rate of computing to communication and different computer nodes on the performance of our simulation framework. In figure 4, we can see that when there is no computing load in computing models. With the number of computer nodes increasing, the time consumed by each loop of all computing models will rise slightly. This is because one thread will be created for each computing model to start it. The increase of the number of threads will reduce the efficiency of communication between a computing model and the collaborative simulation control module, thus influencing the execution time of iteration. When the rate of computing to communication is 3:1, with the increase of the number of computer nodes, computing models run on different nodes in parallel. And the impact of model starting threads on the efficiency of communication is relatively small. Thus the time spending in iteration of each computing model will be decreased. When the rate of computing to communication becomes 30:1, the degree of parallelism for computing models running on more computational nodes will be higher. And the impact of model starting threads on the efficiency of communication can be ignored. Therefore, it can reduce the time consumed by iteration.

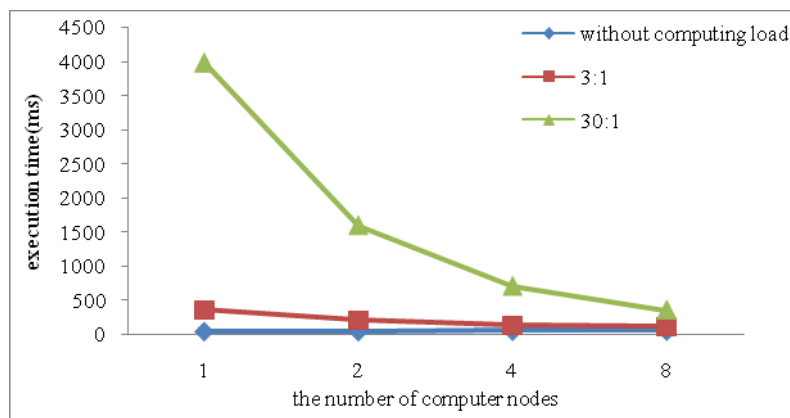


Fig. 4. The performance of different computing rate

The result illustrates that with the rate of computing load to communication load increasing, the parallelism of simulation system model will be higher, thus the performance of the simulation system can be promoted.

4. Conclusion and Future Work

With the development of simulation application, complex system oriented collaborative simulation technology expended from single domain to multidomain. This paper proposed a graphic heterogeneous and

hierarchical modeling approach, and designed a synchronous control algorithm based on synchronous points array. We implemented a collaborative simulation framework based on distributed computing environment. This framework provides an efficient mechanism to realize data sharing among multidisciplinary models. It is an efficient method for complex system simulation. The results of the experiment illustrate that with the rate of computing load to communication load increasing, the parallelism of the simulation system model will be higher, thus the performance can be promoted.

In the future, we plan to design various experiments for our simulation framework on high performance computing environment, and utilize the results to improve it. Another interesting line of investigation would be the research on the efficiency of communication in our collaborative simulation framework.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China under Grant No.61170048.

References

- [1] Richard M. Fujimoto. Parallel and Distributed Simulation Systems. JOHN WILEY&SONS, INC, 2000.
- [2] Xiao Tian-yuan, and Fan Wen-hui. HLA based Integrated Platform for Collaborative Design, Simulation and Optimization [J]. Journal of System Simulation, 2008, 20(13): 3542-3547.
- [3] Maghnouji R, Berbraeck A, et al. Collaborative simulation modeling: experiences and lessons learned. Proceedings of the 34th Hawaii International Conference on System Sciences, Hawaii, U.S.A, 2001, 88-97.
- [4] Van Beek D A. Multi-domain modeling, simulation, and control. Proceedings of 4th International Conference on Automation of Mixed Processes. Dortmund, Germany, 2000, 139-146.
- [5] Miller V T. Hybrid Heterogeneous Hierarchical Model for System Simulation. International Journal in Computer Simulation, Vol 5, 1995.
- [6] Schweiger, Wolfgang; Schoefmann, Werner; Vacca, Andrea. Gerotor Pumps for Automotive Drivetrain Application: a Multi Domain Simulation Approach [C]. SAE International Journal of Passenger Cars Mechanical Systems, v4, n3, p1358-1376, December, 2011.
- [7] Faruque, M.Q.; Dinavahi, V.; Steurer, M.; Monti, A.; Strunz, K.; Martinez, J.A.; Chang, G.W.; Jatskevich, J.; Iravani, R.; Davoudi, A.. Interfacing Issues in Multi-Domain Simulation Tools [J]. IEEE Transactions on Power Delivery, v27, n1, p439-448, January 2012.
- [8] Chai Xudong, Li Bohu, Xiong Guangleng, et al. Research and Implementation on Collaborative Simulation Platform for Complex Product. Computer Integrated Manufacturing Systems. 2002, 8(7):580-584.
- [9] Deng Rui, Zhao Wen, Wang Weiping. Study on Collaborative Modeling Framework for Constructing Distributed Heterogeneous Simulation [J]. Computer Simulation 2005, 22(9):76-80.
- [10] Manuel González, Javier Cuadrado. Mbslab: a New Collaborative Simulation Environment of Multibody System Analysis [J]. Proceedings of ACMD 2004, 383-389.

Contact of Corresponding Author:

Feng Zhu, zhufeng@nudt.edu.cn, +86-13874868244