

Analysis of the Applied Pattern of Distributed Computing Used in Simulation

Song-feng Gao and Hua Qin

Department of Industrial Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, P.R. China

Abstract. This paper discusses about the future applications of distributed computing from the aspect of simulation. It also talks about the concept, classification and application used now of distributed computing. It puts an emphasis on the applied pattern of distributed computing in the field of simulation and comes out with a particular statement on the practice of distributed computing used in the simulating system of human sports biomechanics.

Keywords: Distributed computing, Simulation, .NET Remoting.

1 Introduction

From distributed computing to network computing, from J2EE to .NET Framework, from XML RPC to Web Services, there comes so many conceptions about distributed computing that it's really tough for most technicians and researchers to master them all. This phenomenon is a shadow of the highly developing of computer technology. In spite of this, have all these great concepts and technologies been used widely? Or are they just passing by? How far is the distributed computing away from us, and when can it be really used? This paper presents the application of distributed computing in the field of human sports biomechanics as an example to discuss the problems of distributed computing, hoping that more discusses will come out to promote the developing of the technology of distributed computing.

2 Related Works

2.1 Concept of Distributed Computing

Distributed computing is a science and technology of computer cooperation. It first cut a project which needs lot's of computing into small and easy jobs, then, it distributes the them to many computers separately. When all the computers finish their jobs, the results will be uploaded and combined together. By this way, the complex project is finished[1].

2.2 Classification of Distributed Computing

There are many ways to classify distributed computing.

From the aspect of net, it can be classified as Local Area Network Distributed Computing and Wide Area Network Distributed Computing[2]. The main difference between them is where the nodes lie---in Local Area Network or in Wide Area Network. Another difference of them is the protocol they used. In Wide Area Network, the protocol of Http is used while in Local Area Network TCP/IP is used.

From the framework of application system, it can be classified as follows:

Firstly, the J2EE system. The developing language is in Java. It has an standing out advantage --- independence on platforms. It needs the JVM as the runtime environment. Since it is a translated language, the efficiency is somewhat lost.

Secondly, the .Net system. It is a distributed system supplied by the Microsoft Co. It is quite similar to the J2EE system. All the developing language from the Microsoft Co. is supported in this framework, including C#, VC.NET, VB.NET, etc. It needs the .Net Framework as the runtime environment. What's more, the operating system should be Windows 2000/XP and later systems.

Thirdly, the CORBA system. It's the distributed system supplied by the organization of OMG. It provides the application developing rules instead of details. From this point, we can use any language to develop a COBRA system. However, since it's hard to master it, success applications of COBRA are rather few by now.

2.3 Application Situation of Distributed Computing

So far, most distributed computing projects make full use of the volunteer computers' capability which has been left unused. These computers communicate by Internet from different places in the world.

A project named SETI@home is the biggest project of distributed computing. It analyses the radio signal from outer space in order to find the evidence of life there. The data it contains are quite huge ---more than ten million digits. Four million computers have joined this project (about 40 thousand volunteers in mainland of China).

Another project called Folding@home analyses the inner structure of protein and drugs related. More than 100 thousand volunteers have joined (about 60volunteers in mainland of China). It is such a huge project that the amount of calculation is striking and it is impossible for a single computer to carry it out.

The application of distributed computing is not common so far. It still stays at the phase of research and doesn't come to the step of really use. With the highly developing of hardware, the ability of calculating has greatly improved. For single applications, it is already enough. However, in fact, most huge projects of today are not single applications; they are combination of many single applications, usually called a team or a group. So, the problem of distributed computing pattern should be solved from application pattern rather than concepts.

3 Methodology

3.1 Analysis of Application Situation of Distributed Computing in Simulation

Distributed computing has not been really applied in simulation yet. The existing mature software for simulation does not contain any pattern of distributed computing. That's to say, these software such as Proe, Ansys, Adams are all stand-alone version and do not have the concept of distributed computing at all. It takes a very long time to develop this software. At the very beginning of designation, there exists no concept of distributed computing at all. For these two reasons, it is impossible to add the concept of distributed computing into this software. Then, is there any possibility of using this concept in simulation? The answer is yes. There is a bright future for the application of distributed computing in the field of simulation.

Why saying that? First of all, software for simulation are all complicated and huge ones. The requirements on hardware are rigid. Generally speaking, a personal PC (not a workstation) can run simulating software only at same time. In this case, if we want to run more than one simulating software at same time or if we want to run more than one examples in one simulating software, we need more than one computer to cooperate. Second, the existing software for simulation does not support the mode of distributed computing. So, we have to load the computing task step by step while not computing parallel. Today, the hardware level of personal PC has already exceeded that of a workstation years ago. Why not use the resource left unused of the computers in the network of local area or wide area?

So, if we can gather the resource of computer available in local area and wide area to form an environment for simulating, we can not only make full use of our computers without any need to increase hardware but also optimizing our developing of software for simulation. In this way, every node can be developed step by step separately. At the same time, we can build the developing mode by cooperating with other companies easily in this distributed developing environment.

In a word, by building the mode of distributed computing, we can optimize both the software and hardware environment. We can say that, applying the concept of distributed computing in the field of simulation is not only feasible but more than feasible.

3.2 Carry Out Distributed Computing Using .NET Remoting

Most of the operating system of personal PC is Windows with the version of 2000+, and most of the software for simulation has a version for windows. Each system of J2EE, COBRA and .NET has its own advantages and disadvantages. J2EE is platform independent, but its efficiency is low. The worst of all is that most software for simulation do not support the API for java. COBRA is common, but it takes time to master its technology. It is no easy. Only .NET satisfies us. It is not the key problem that it doesn't have the ability to cross platforms. For the reasons above, we choose .NET to carry out the distributed computing for simulation.

NET Remoting is a new concept born in the .Net Framework. It's a further development of COM+. It supplies a framework which allows objects to influence each other through applications. This framework supplies many services including

activating and lifetime support together with communicating channels for transferring messages with remote applications.

NET Remoting supplies two communicating channels: HTTP and TCP. This means that we can not only construct a application of distributed simulation by TCP channels in a local area network but also distribute our node for distributed application by HTTP in the wide area network.

HTTP channels use the protocol of SOAP to transfer messages with a remote object. When the messages go through the SOAP-Formatting application, they are formatted into XML and serialized. TCP channels use a Binary-Formatting application to format the messages into a binary stream and transfer them to the target URL by the protocol of TCP.

When we use the .Net Remoting to construct an application of distributed computing for simulation, we should first figure out how the simulating system should be distributed in the network environment (local area network or wide area network). In details, where the node should be distributed and what is the relationship between them.

Then, we should build the control center of the distributed computing system. The control center is responsible for controlling the work of each node including starting and stopping the application, the control of action, importing and exporting data, working procedure arrangement and executing, dynamic distribution and adjustment of nodes, etc.

What's more, we should install corresponding control service on each node so that each node can receive the commands from the control center and feeds back its results.

3.3 Application of Distributed Computing in the Simulation of Human Sports Biomechanics

In the practices of building our human sports biomechanics simulating system, we make full use of the concept of distributed computing. By this way, all the nodes unite closely. Under the control of controlling center, it makes full use of the computer resource available in the local area network, succeeding in binding Adams and Ansys closely together so can influence each other. This does realize the integration of the application of distributed computing simulation. The picture for this model is as follows:

This system is based on .NET Framework. The developing language is C# and VC++.NET. The key distributed technology is by .NET Remoting.

During the process of building the system, we focus on these problems:

✧ Layout of the nodes

This relates to two problems. The first one is the problem of hardware. We figure out the distribution of application according to the computer resource in the local area network. The second one is the software problem. According to the plot of the nodes, a different person is responsible for corresponding simulating application[3]. By this way, each node is developed separately and distributed dynamically.

✧ Carry out of node service

Each node is corresponding to a client of the control center in fact. It follows the commands from control center and acts. The precondition for receiving the

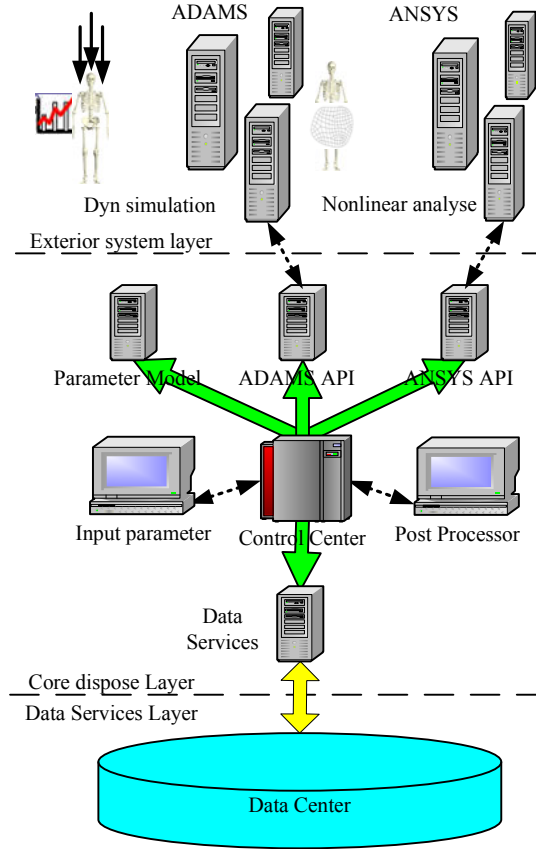


Fig.1. Human Dyn Simulation System Framework

commands is that the service of the node should be started first. This service is the key of the distributed computing system.

There are many forms to build such a service of the node. Here we choose the NT Service to construct our stand windows service. When the service starts, HTTP channel and TCP channel of .NET Remoting are registered. At the same time, the service for receiving commands form control center is bind. Therefore, when this service of the node is started, the node will connect to the control center automatically.

◇ Further developing of the application of nodes

There distributes different kinds of application in each node, which should be further developed so that it could be accepted by the distributed computing system. This is because that all the existing software for simulation are in stand-alone mode and they finish all the procedure in one computer. Further developing of the application of node is to realize the target that the software for simulating and control center can act to influence each other by the supplied further developing tools and

corresponding node service. For example, we can use the APIs supplied by Adams to build a connection between Adams and the service of the node[4]. By using the methods the .NET Remoting we can call the service of the HTTP channel and TCP channel so as to connect the service of the node with the control center. By this way, we make Adams able to receive the commands from the control center and feed the result back.

✧ Communication between nodes

The communication between nodes is realized by the control center in principle so as to avoid direct communicating. The main benefit of doing so is that we can avoid the reticulate structure of the software and make sure it is a star structure. It is good for the the system to expand and maintain. In detail, the communication between node A and node B goes in this way: “node A -----control center-----node B”.

✧ Constructing control center

Control center is the heart of the system. All the commands come out from it and all the working procedures are figured out here. Control center manipulates from the input of parameters to the building of concrete model for human body, to the building of dynamic model, to the building of partly nonlinear model and later processing model[5]. It is consisted of the following components:

- [1] Component Function Trigger: Controller to trigger all the function of the components.
- [2] Work Flow Marshal: Configure the work flow of the system.
- [3] Distributed Computing Marshal: Configure the distribution of the components.
- [4] System Parameter Marshal: Configure the system's runtime parameters.
- [5] System Monitor: Monitor the situation when system is running.

Above content presents key steps when developing a distributed computing human sports biomechanics simulating system. It takes quite a long time to develop such a system and needs a software developing mode of waterfall. It accumulates and updates little by little and day by day. We should follow this rule when developing this system so as to guarantee for success.

4 Results

It is sure that we can build a distributed computing application in the field of simulation. There are three modes to build it: .NET、J2EE、CORBA. The best is to use the .NET framework. Use the .NET Remoting to build the communicating channel and inter-invoke mechanism. The practice of the distributed computing human sports biomechanics simulating system proves that: As a result of the developing of hardware and software, the field of simulation will surely develop into a distributed structure.

5 Discussion and Conclusion

This work was supported by the basal scientific research plan of Commission of science technology and industry for national defense.

References

1. Banerjee, A., Corera, A.: C# Web Services-Building Web Services with .NET Remoting and ASP.NET. Wrox Press, USA (2004)
2. Obermeyer, P., Hawkins, J.: Microsoft.NET Remoting summarize. Wrox Press, USA (2003)
3. Zheng, J.-R.: ADAMS dummy Product accident and enhance. Machine industry press, Beijing (2002)
4. Liang, H.: A Dynamic Simulation System for Flexible Robot Manipulators Based on ADAMS & ANSYS. Mechanical Science and Technology 21(6), 205–208 (2002)
5. Liu, Z.-J., Wu, Y., Huang, S.N.: Research of Dynamical Simulation in High Speed Cigarette-cutting Support System. Journal of Hunan University (Natural Sciences Edition) 29(6), 132–135 (2002)