Design of Distributed Real-time Simulation System Based on RTX and Reflective Memory Network

Pei-yao Li* and Wen-guang Li

School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

*Corresponding author

Keywords: RTX, reflective memory, real-time communication protocol, distributed real-time.

Abstract. The hardware in the-loop simulation is an important method to speed up the research process of spacecraft, and the real time simulation system is the core of the hardware in the loop simulation. Distributed real-time simulation system based on RTX and reflective memory network is upgrading and expansion from the EMES simulation system, mainly expansion on the simulation calculation, simulation scheduling and network communication, and design the real time scheduling algorithm based on RTX high precision timer and communication protocol based on reflective memory network, the simulation system using dual network transmission technology. At last, the experimental results show that, the system is real-time, and the communication is stable, high speed and reliable, and the simulation system meets the requirements of real-time simulation.

Introduction

The simulation technology is based on the technical expertise of similarity, information technology, system theory and the relevant applications in the fields of technology [1]. As the simulation technology is going to be distributed and in real time, it will be the focus of the research of the semi-object distributed real-time simulation system. The existing EMES simulation system is a non-real-time simulation system based on Windows and Ethernet. Because of the large interrupt delay and uncertain thread scheduling mechanism of Windows system, and the low clock resolution make the simulation system based on Windows operating system unable to meet the requirements of real-time determination. Moreover, because Ethernet uses CSMA/CD mechanism to control data transmission, the delay of data transmission is uncertain. Each network transport endpoint has to compete to obtain the right to send packets. Based on the above shortcomings of EMES simulation system, this paper designs a distributed real-time simulation system based on RTX and reflective memory network by using RTX and reflective memory network to improve the simulation calculation, scheduling and the speed of network communication.

The Architecture of the Simulation System

The real-time simulation system is a distributed real-time simulation system, which includes the main control terminal that in charge of simulation scheduling and management, the computing node terminal distributed on each simulation computer, the simulation database and the simulation object, etc. Each subsystem software uses Ethernet and reflective memory network for network communication and data exchange. The main simulation control terminal is the core of the distributed real-time simulation system, including several functional modules, such as simulation model management, simulation test management, simulation test running scheduling, simulation data playback and evaluation etc. The simulation node machine and the simulation calculation program adopt the shared memory to carry on the data interaction, in addition, the hardware-in-the-loop simulation equipment can be connected into the simulation node machine, the simulation database stores the simulation model, the simulation experiment, the model file and the information of simulation data file.
Design of RTX Real-Time Extension Module

RTX kernel uses preemptive and time slice rotation algorithms to achieve high speed thread scheduling. To keep thread switching time in the range of 0.5 to 2 microseconds, the timer cycle interval can achieve 100 microsecond RTX memory allocation in the non-paging memory pool, it avoids the delay caused by missing pages in the use of virtual memory. Thus, the deterministic RTX for external event response can provide a stable and high performance real-time extension to the kernel level of Windows, and supports the sustained interrupt triggering speed of 30KHZ and the typical interrupt delay of Core2Duo system is less than 10 microseconds [2].

This article will add RTX real-time simulation module into EMES simulation system, the simulation control terminal and compute nodes are joined RTX real-time simulation module, and then the simulation calculation program and other data files sent to the corresponding simulation nodes, starting the simulation program before completing the real-time simulation of initialization operation, waiting for the user to start the simulation. Distributed real-time simulation system of real-time simulation module and the original simulation module are independent of each other, the real-time simulation program in RTX real-time simulation module calculation, both the event and the shared memory for data synchronization, we would discuss in details in this paper with the followings such as the interface from the simulation model and real-time simulation program to generate interaction, real-time simulation process and the real time clock, etc.

Real-time Simulation Process

Real-time simulation runs with RTX extension module, real-time simulation program running in RTX environment is independent of the simulation system, and the shared memory and event mechanism are used for synchronization and data interaction. When to start the simulation, simulation system reads the simulation test configuration and file information, and then carries on the simulation node distribution, simulation file is generated, create a shared memory, start the RTSS process of real-time simulation. When the RTSS process of real-time simulation receives the started simulation command, RTSS process started by the timer loop body run every frame simulation tasks, at the same time in each frame of the simulation task after the calculation and simulation status updates by Shared memory data communication with the Win32 process. If the simulation task timed out, record the current timeout frame. The process control of RTX environment and simulation environment under Windows in simulation test is shown in Figure 1.
Real-time Simulation Data Interaction.

In a distributed real-time simulation system, the simulation process can be divided into Windows RTX real-time and non real-time process, the event data simulation and simulation state interaction among the two process through Shared memory. RTX provides RtCreateSharedMemory() and RtOpenSharedMemory() functions to create and open Shared memory for data interaction between the Windows process and the RTX process. Shared memory exists in RTX non-paged physical memory and is passed through Shared memory names. After the simulation, RtCloseHandel() is used to release the original Shared memory.

When the simulation is initialized, a Shared memory is created by the Windows process and locked, and the Shared memory is opened in the RTX project to perform the data interaction of Shared memory.
Real-time Scheduling Driver Clock

Because of the system timer of Windows operating resolution is low, it can only reach the level of millisecond, and the delay timer is bigger, so the Windows operating system is difficult to achieve accurate simulation scheduling, nor cannot meet the requirements of real-time simulation. RTX real-time extension system provides a high precision accurate clocks and timers have made it possible to real-time scheduling, RT - HAL clock can be accurate to 1 microsecond, clock resolution can reach 100 nanoseconds, interval timer cycle can reach 100, 200, 500, 1000 microseconds.

RTX provides a rich RT-API function, in which RtCreateTimer() can be used to create a high precision timer, RtSetTimerRelative() can be used to set the timer count time interval, and the accuracy can reach 10-7s. The high precision time control can be achieved by putting a frame of simulation task into the clock trigger function. The sample code for the timer setting is as follows:

```c
//Create clock
hTimer=RtCreateTimer(NULL,0,TimerHandler,NULL,RT_PRIORITY_MAX,CLOCK)
// Set the clock interval, the frame time.
RtSetTimerRelative( hTimer, &liPeriod, &liPeriod)
//Set the clock trigger function.
voidRTFCNDCLTimerHandler(PVOIDcontext)
{
    Simulation task calculation;
    Simulation scheduling;
}
```

The simulation frame time liPeriod and the simulation total time SimEndTime are determined by the user according to the actual situation.

Design of Communication Network

Ethernet uses the CSMA/CD mechanism for transmission control with the defects of data delay uncertainty. In addition, after the channel information start sending, communication machine also need to check whether the data conflict, if the data conflict, it also need to send data, because of data conflict cannot ensure communication response, real-time Ethernet cannot meet the strict requirements on the real-time simulation in real time.

In comparison, this paper uses PCI - 5565 reflective memory card. When a node machine after writing data, all data will be rapidly through the optical fiber transmission to the network interface card memory, optical network transmission serial baud rate can be up to 2.12 G, and the process was conducted by the on-board circuit automatically, without the CPU of a node machine. This significantly save the CPU resources for data calculation [3].

Design of the Reflection Memory Network Communication Protocol

Compared with the traditional Ethernet, reflective memory network protocol is simple, easy to use, and can read and write data by calling the API function directly, it needs to write different read and write operations according to different application refer to the TCP/IP communication protocol, this paper designed reflective memory network communication protocol stack, meeting the requirements of distributed real-time simulation system. The overall structure of the reflection memory network communication protocol is shown in Figure 2.
Setting the card ID of the various memory card as the physical address of reflective memory network, using ARP protocol to map the network physical address and IP address, for more complex TCP protocol, using object-oriented method, encapsulates the two class of TCP server and client, and realizing the basic methods such as Listen, Connect and Bind, Send and Receive, calling IP module to Send and Receive message.

**Reflective Memory Space Partitioning**

Distributed real-time simulation system has high real-time demand of data, data reading and writing frequency is fast, if by setting the flag bits for each node to coordinate, speaking, reading and writing will reduce the bandwidth of reflective memory network, this cannot meet the requirement of real-time, so you need to uniform the reflective memory card storage space allocation and management, distributing a enough memory space of each node, when the master need different node information, it can read data to the corresponding node machine reflective memory offset, this improves the bandwidth of the reflective memory network, avoids the data conflict and coverage. According to the research on the real-time simulation system, the node machine transmission of data size is commonly a few KB to dozens of KB, the largest amount of data for the level of KB, as a result, each node pre-allocated 1 MB of storage space.

**Comparison of Simulation Results**

This section validates the real-time of distributed real-time simulation system by constructing a concrete example, comparative analysis on the Ethernet and reflective memory network, the difference in the real-time simulation between the Windows system and RTX real-time, the high precision timer module was integrated into the simulation system, so the simulation time of the simulation process can be recorded. The simulation test runs 840 steps in total, and the communication time of each step is recorded, and the data analysis is performed using MATLAB.

**Comparison Analysis of Emulation Communication Time between Ethernet and Reflective Memory Network**

This section is designed to test the communication time, analysis the difference of Ethernet and reflective memory network about communication time separately, Ethernet and reflective memory network data transmission time contrast figure and table are shown in Figure 3 and TABLE 1, the curve in red represents the reflective memory network and the grey is the Ethernet, we can obviously found that the average time of reflective memory network communication is half of the Ethernet by contrast, the memory network has high speed, low latency. This illustrates the stability of the reflected memory network communication protocol as designed herein. Through the Figure 3,
we obviously found that reflective memory network communication time is stable, the curve has been in a period of very small range. On the contrary, the range of Ethernet communication time is great, the largest average time consuming nearly six times more than the time-consuming, this influence is too big in the real-time simulation system. It can't satisfy the needs of real-time either. Therefore, in the strong real-time simulation system, reflective memory network or other real-time network with strong real-time, low latency must be used.

![Figure 3. The contrast of Ethernet and reflective memory network in communication time.](image)

<table>
<thead>
<tr>
<th>Scene</th>
<th>Min(ms)</th>
<th>Max(ms)</th>
<th>Average(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>0.33</td>
<td>2.55</td>
<td>0.41</td>
</tr>
<tr>
<td>Reflective memory</td>
<td>0.17</td>
<td>0.31</td>
<td>0.21</td>
</tr>
</tbody>
</table>

**Table 1. The contrast of Ethernet and reflective memory network in communication time.**

Comparison and Analysis of Simulation Time between Ethernet and Reflective Memory Network Communication in Windows Environment

In this section, the simulation test is carried out in the Windows environment, and the data transmission in Ethernet and reflection memory network are FIGURE 4 and TABLE 2 are the total time consuming comparison charts and comparison tables for the simulation of communication between Ethernet and reflection memory network respectively, the curve in red represents the reflective memory network and the grey is the Ethernet.

Through the contrast, the data communication time is stable in reflective memory, but the simulation time-consuming under Windows still volatile, by contrast we can see the average time of network data communication is about 0.2 millisecond, the total length of the trial was 2.5 millisecond., test running and scheduling in the process of time-consuming is larger, a time-consuming preliminary speculation over the cause of the volatile is to test run time-consuming and scheduling time uncertainty. On this basis, this paper design of Windows and using RTX environment reflective memory network simulation time contrast test, the network communication are all in the reflective memory, and test the time consuming under the environment of Windows with RTX.
Summary

This paper takes the simulation system under Windows as the research object, real time simulation system through RTX extension under Windows simulation system, the reflection memory network is introduced to enhance the real-time performance of real-time in data transmission of the simulation system. Through example validation, it can be shown that the real-time and practicality of the real-time simulation system based on RTX and reflective memory network meets the desired goal, it provides a favorable support for the research of real-time simulation system. It is also beneficial to the construction of high performance real-time simulation system.

References

