REAL-TIME, DISTRIBUTED, PARALLEL SIMULATION OF HYDRALIC SYSTEM BASED ON LINUX IN THE NETWORK

Li Kun and Jiao Zongxia

College of Automation Science and Electrical Engineering
Beijing University of Aeronautics and Astronautics
Beijing 100083 P. R. China

Keywords: *Hydraulic system, Parallel simulation, UDP/IP protocol*

Abstract:

For some real-time and complex hydraulic system, the normal computer cannot meet with the system requirement. Consequently the parallel simulation would make a good business. It can reduce the cost of hardware and enhance the speed of simulation, and realize the real-time simulation of the complex hydraulic system.

This paper discusses the real-time, distributed, parallel simulation of hydraulic system based on Linux in the network. Linux have power ability in the network, which is very stable and have little requirement of hardware[1]. So it is worth studying in hydraulic system simulation and control. According to the complex degree, the number of computers in network is chosen. According to the real-time system features such as frequently changing data which are small every time, UDP/IP protocol is chosen. However, the most serious defect of UDP/IP protocol is its unreliability because it is not based on connection and cannot guarantee the credibility of the data transmission. Thereby, the signal system of Linux is chosen to prevent from losing data by judging the successfully sending-data and avoid UDP jam. So the accuracy and reliability is ensured. The Runge-Kutta is chosen to solve the state equation and realize the parallel calculating.

1 Preface

There are some characteristics of hydraulic system simulation which is different from other control systems simulation. The regulating effect of pump and the long pipe must be considered. Then, the system would be very complex and the state equation of the system would be more than one hundred orders. So the enormous calculating would make a normal computer calculate for a long time. Distributed and parallel simulation will meet the requirement.

Nowadays Linux develop very fast .It is a free operating system and it is very stable and have little requirement of hardware. It supply multi-processes, so the user can establish multi-processes to finish his work. Moreover it have powerful ability in the network, which supply TCP/IP, UDP/IP, SLIP and PPP protocol. Especially it supply socket program so users can programme easily[1]. So it is worth studying in hydraulic system simulation and control.

2 The parallel simulation based on Linux

2.1 Communication protocol

TCP/IP is based on connection. It will not send one word unless it received destination's reply. So it can guarantee the credibility of the data transmission. UDP/IP is not based on connection so it cannot guarantee the credibility of the data transmission but it has a fast transmission rate[2]. Because the real-time system change data frequently, UDP/IP protocol is chosen.

Two computers are chosen in a local area network which have ten computers connected by a 10M HUB. One is client and another is a server. The client send 64 bytes to the server. The server send data to client immediately after it receive the data. By the way the transmission time is measured. The tests show that 29-32μs is cost by using UDP/IP protocol and 140-180μs is cost by using TCP/IP protocol. So we can draw a conclusion that the transmission rate of UDP/IP is much faster than TCP/IP's.

UDP/IP cannot guarantee the credibility of the data transmission. In order to know exactly the probability of its losing data package, a program is designed to test. Using UDP/IP protocol the client sends a data to server. After receive the data, the server adds 1 to the data and sends back the client. Then the client does the same work as server adding 1 to the data and sends back. The circulation are done 200,000,000 times and every transmission successfully. So we can draw another conclusion that even though the UDP/IP cannot guarantee the credibility of the data transmission, its probability of losing data package is very minute.

2.2 credibility realization

The program should guarantee the credibility of the data transmission using UDP/IP protocol because UDP/IP never ask for reply after sending data, so it doesn't know whether the data transmission successfully.

There is signal system in Linux which is similar to hardware interruption. Linux defines some signals for running processes' software and hardware state. The signal system is called "software interruption" which can be sent to another process by one process[3].

So the signal system is used to guarantee the credibility of the data transmission. When the data is sent by UDP/IP, the function of "alarm()" is set to count. If the data lost and no any reply are received in limited time, the function will sends a signal to interrupt the program and make the lost data send again. And then the credibility of data transmission is realized.

2.3 Server-client model

According to the server has to deal with several clients, the server creates several son-processes to deal with clients. When the server receives one client's request, it selects a new and free port and creates a son- process which communicates the client with the new port. And the father- process keep on waiting other clients request. Using the server-client model, many clients' request can be done by different son-processes in the server. As follow figure:

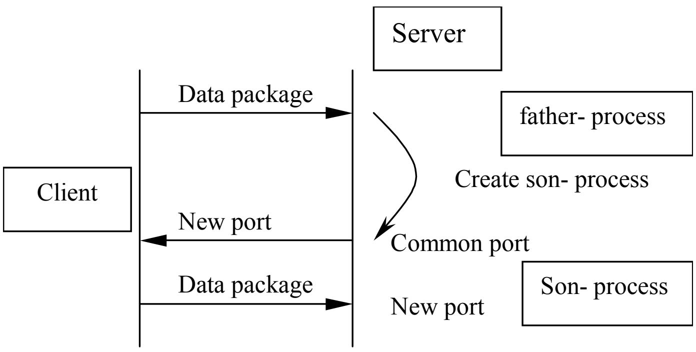

Figure 1 server-client model

3 Distributed, Parallel Simulation of hydraulic system

3.1 Hydraulic servo system model

Hydraulic servo system model is composed of controlling component and executing component. The controlling component is hydraulic servo valve and the executing component is valve control motor. All pipes are supposed to be short and wide. The friction in the pipe and the weight of liquid can be ignored. At the same time the temperature and Young's modulus of oil are assumed constant. At last the four system equations are gotten as follows[4]:

（1）controlling component equation:

$$Q_f = CWx_V\sqrt{P_s - P_f / \rho} \qquad (1)$$

（2）flux succession equation:

$$Q_f = D_m d\theta \ / dt + VdP_f /(4E_h dt) + k_{tc}P_f \qquad (2)$$

（3）motor moment balance equation:

$$M = D_m p_f = J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + G(\theta - \theta_f) + T_L \qquad (3)$$

（4）servo valve mandril shift equation:

$$x_v = K_{is}K_r(\theta_s - \theta_m) \qquad (4)$$

variable substitution：

$$Y = \dot{\theta} \qquad (5)$$

$$\dot{P}_f = \frac{4E_h}{V}[CWK_{is}K_r(\theta_s - \theta_m)\sqrt{\frac{P_s - P_f}{\rho}} - D_m Y - k_{tc}P_f \qquad (6)$$

$$\dot{Y} = \frac{1}{J}(D_m p_f - BY - G\theta_m - T_L) \qquad (7)$$

3.2 The pump system model

The constant pressure pumps always are used in hydraulic system to meet the request of constant pressure. There are some features in this system such as response quick and low excursion. The state equation of the system is as follow[4]:

(8) overflow valve flux succession equation:

$$Q_e = CW_2 x\sqrt{2P_s / \rho} \quad (x>=0) \qquad (8)$$

$$Q_e = 0 \qquad (x<0) \qquad (9)$$

$$P_s A = kx_0 + kx + B_e\dot{x} + m\ddot{x} \qquad (10)$$

variable substitution:

$$Y = \dot{x} \qquad (11)$$

$$\dot{P}_s = \frac{E_y}{V}(Q_s - CWx\sqrt{\frac{2P_s}{\rho}} - Q_f) \qquad (12)$$

$$\dot{Y} = (P_s A - kx_0 - k - B_e\dot{x}) / m \qquad (13)$$

3.3 Parallel simulation

The server-client model is used to realize the system parallel simulation. The pump system is server and the servo system is client. Runge-Kutta function is used to calculate the differential eguations and realize the parallel simulation.

Two variable arrays are defined in the two computers: Q_f=buf[0], Ps=buf[1]. First, the servo system sends one data Q_f=0 to the pump system which is server. After received the data, the pump system model just have three variables in three equations and begin use the Runge-Kutta function to calculate the differential eguations. At the same time it sends one data Ps to the servo system which is as the client. In the same way, there are only three variables in the three differential eguations and begin to calculate. After the Runge-Kutta finished the calculation, the server and the client exchange the data and begin the second calculate. Through this way, they exchange data n times and calculate the models. With the data which is calculated at every time, the figures of the two models can be drawn. Flow chart is as follow.

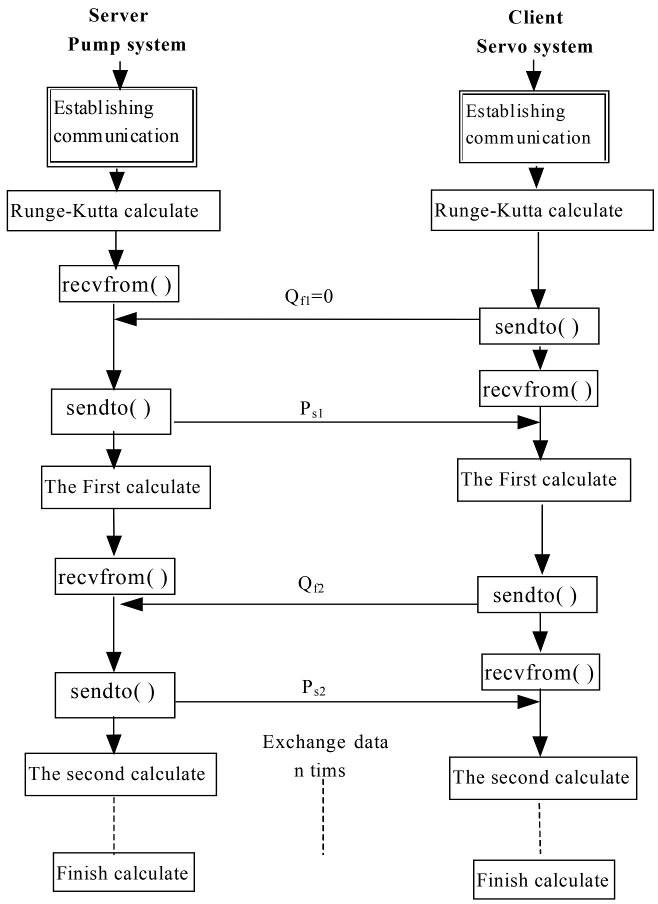

Figure2 Flow chart

By this means, the real-time, distributed, parallel simulation of hydraulic system based on Linux in the network is realized. The simulation result tells us that there is wonderful real-time of distributed, parallel simulation in Linux. The model which is chosen is perfect, to most real systems which are more complicated the distributed the parallel simulation will has much advantage.

4 Summarize

Because of the Linux advantage such as: high efficiency, stable, little requirement of hardware and supplying multi-processes etc, the distributed, parallel simulation in Linux can be used to design the complex hydraulic system. The most important problem: communication in network and parallel calculate between processes are resolved commendably.

References

[1] Lin Yu. *Network program in Linux*. The people post publisher, 2000,10.

[2] Liu Peizhi. Computer simulation. *The research of communication in real-time network in distributed simulation*. Vol.5,2000,9.

[3] John Goerzen. *Programming in Linux*. The electronic publisher, 2000,12

[4] Wang Zhanlin. *Servocontrol of hydraulic*. The Beijing University of Aeronautics and Astronautics's publisher, 1987,2