

Research on the Application of Agent-based Real-time Monitoring System in Inference Engine Cluster

Xu Jiangtao

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1400693814@qq.com

Yang Bo

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 16764496@qq.com

Liu Pingping

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 134369601@qq.com

Abstract—In order to ensure the stable, reliable and healthy operation of spacecraft, this paper introduces the spacecraft expert diagnosis system for real-time monitoring and early warning of spacecraft. The inference engine is the core of the spacecraft expert system. Each inference engine is responsible for a spacecraft monitoring and early warning. How to ensure the stable and reliable operation of the inference engine is an urgent problem to be solved. In this paper, agent technology is introduced to monitor the reliable operation of the inference engine and process migration when the inference engine fails. At the same time, monitor server resource utilization for process scheduling to achieve server load balancing. Practice has proved that the system can greatly improve the stability of the inference engine, improve the efficiency of spacecraft management, and save a lot of manpower and material resources.

Keywords-Multi-reasoning Machine; Process Scheduling; Reliability; Agent; Monitoring

I INTRODUCTION

With the rapid development of China's aerospace industry, the monitoring and management of spacecraft has become more and more important. Spacecraft monitoring parameters as few as five or six hundred, as many as several thousand, each parameter anomaly may affect the normal operation of the spacecraft system, and

even lead to the paralysis of the spacecraft, thus bringing immeasurable losses to the country. In the traditional management of spacecraft, the spacecraft managers monitor the operation parameters of the spacecraft, and send them to the spacecraft experts after finding the abnormal parameters. The spacecraft experts analyze the abnormal parameters, and then give the treatment scheme. Due to the small number of spacecraft, it can better complete the spacecraft monitoring and management tasks. However, with the increase in the number of spacecrafts, the mode of manual monitoring and analysis can no longer meet the needs of spacecraft monitoring and management.

So the spacecraft expert diagnosis system [2] [4] [5] platform is introduced. The knowledge of the spacecraft expert [3] is expressed as the knowledge that the inference engine can handle. The knowledge is loaded when the inference engine [1] runs, and the diagnosis results are given combined with the current state of the spacecraft operation parameters. In the specific implementation, each inference engine is responsible for real-time processing of the operating parameters of a spacecraft, so that all spacecrafts form a cluster of inference engines [6] [7]. Obviously, the stable operation of the inference engine is directly related to the safety of

the spacecraft. So, how to make these inference engine cluster reliable and stable operation is an urgent problem to be solved, this paper gives practical and feasible solutions to this problem. Practice has proved that the system can not only greatly improve the stability of the inference engine, but also optimize the system resources and dynamically schedule the inference engine process, thus greatly improving the efficiency of spacecraft management and saving a lot of manpower and material resources.

At present, major aerospace powers are carrying out research on spacecraft fault management technology. NASA has made spacecraft failure technology the first requirement of space flight technology in the 21st century. NASA's spacecraft fault management projects include X-33 / X-34 / X-37, and military aircraft projects include F-18, F-22, JSF, UCAV, etc. The fault management technology in the United States mainly has two representatives: 1 integrated fault management technology (IVHM) for carriers, 2 prediction and fault management technology (PHM). At present, there is still a big gap between our country's research on spacecraft fault management technology and world technology, and there is no comprehensive research and verification of spacecraft fault management technology. On the one hand, our overall scientific

and technological level is lower than that of developed countries. On the other hand, because of the small number of spacecrafts, the current ground measurement and control system can still be maintained. However, with the increase in the number of spacecrafts, the contradiction of low management level highlights. How to solve this contradiction and improve the management level of spacecrafts by using spacecraft fault knowledge management technology has become the only way for us.

II SYSTEM STRUCTURE

According to the characteristics of inference engine cluster real-time monitoring software running environment, this paper adopts the following system structure.

A. Network Topology Architecture

The spacecraft expert diagnosis system platform [8] [9] is running on the dedicated network. Including: several servers, several clients and database servers. The inference engine process runs on the server. In order to ensure the stable operation of the inference engine process, in general, a server installs a inference engine process and runs a inference engine process. The network topology is shown in Figure 1.

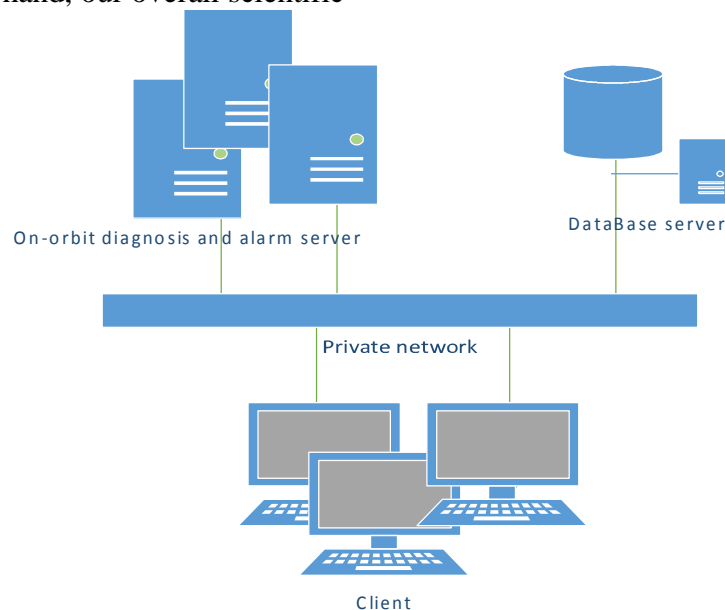


Figure 1. Topology Structure Diagram of Private Network

When multiple servers participate in the actual work, the main monitor disperses the huge data processing business to each server to achieve load balancing, so that multiple servers process the corresponding data at the same time, and thus can be competent for a large number of data processing tasks under the new demand. In each server mode in the LAN, even if the amount of data information increases again later, it only needs to be solved by task scheduling method or increasing the number of servers.

B. Factors Affecting Scheduling Structure

To realize the multi-satellite fault diagnosis task division, the scheduling structure must be formulated firstly. The scheduling structure is the basis of designing the optimization model and solving algorithm. The scheduling structure of the multi-satellite fault diagnosis system must fully consider the characteristics of the problem and the requirements of the system in terms of scalability and maintainability, including:

1) Constraint Representation of Satellite Telemetry Data and Processing

Single star fault diagnosis task planning is only for a star telemetry data analysis. When there are multiple satellites, due to different satellite data formats and different time constraints on telemetry data analysis, it must be completed within the specified time, otherwise it will lead to disastrous consequences. It is difficult to uniformly represent the constraints of multi-satellite fault diagnosis system. Only some important constraints are considered, and some secondary constraints are ignored.

2) System Scalability

With the continuous development of satellite performance and quantity, multi-satellite fault diagnosis system task planning must have good scalability to facilitate the dynamic loading and reconstruction of satellite fault diagnosis system.

3) Efficiency and Effect of Scheduling Algorithm

The task scheduling of multi-satellite fault diagnosis system has the characteristics of combinatorial optimization problem, which has

been proved to be NP-hard problem. When the number of tasks and satellites increases, it is very difficult to solve the problem. Multi - star scheduling structure must consider the design of optimization algorithm, so that it can obtain satisfactory solutions with certain quality in a certain time range. In addition, the system should also consider factors such as cost and maintainability.

C. Comparison of Scheduling Structure

Aiming at the task scheduling problem of multi-satellite fault diagnosis system, this paper studies the scheduling structure of multi-satellite fault diagnosis. At present, there are mainly two scheduling structures: centralized scheduling structure and hierarchical scheduling structure.

1) Centralized Scheduling Structure

Centralized scheduling structure: There is a scheduling host responsible for collecting system load information in the system, which is the main body of centralized scheduling. It controls the solution and keeps synchronization with the information interaction between multiple computing hosts. The host maintains a task allocation table and assigns tasks according to the system load. Other hosts are computational hosts, and computational hosts are only responsible for processing telemetry data.

The advantages of this strategy are: the scheduling host has global information, easy to make decisions and maintain load balance, easy to track execution. The algorithm is easy to implement, suitable for the network environment with fewer nodes, and has better performance on bus network. The disadvantage of this strategy is that the computing host needs to wait when scheduling the host to collect information of all hosts, which wastes the processing power of the computing host.

2) Hierarchical Scheduling Structure

The core idea of the hierarchical scheduling structure is that the task pre-scheduling center allocates tasks to different scheduling modules according to the amount of satellite telemetry data. Each scheduling module handles the fault diagnosis of one satellite, and then each

scheduling module allocates tasks to different servers.

The advantage of this strategy is that the multi-satellite fault diagnosis task scheduling is transformed into multiple scheduling module problems by pre-allocation of tasks, which reduces the difficulty of solving the problem. Since each scheduling module only conducts fault diagnosis task scheduling for one satellite, the problem scale is small, and the fault diagnosis task scheduling of multiple satellites can be operated in parallel, so the efficiency of problem solving is high. The downside is that additional scheduling hosts are needed to increase costs.

D. Scheduling Structure Selection

In order to meet the scalability and solution requirements of multi-satellite fault diagnosis system, this paper proposes a distributed

scheduling structure. It includes core scheduling modules and extensible modules.

1) Core Scheduling Module

The subject of centralized scheduling performs solution control and information interaction with multiple extended modules and maintains synchronization.

2) Extensible Module

It is responsible for fault diagnosis of telemetry data of multiple satellites. When there are additional satellites and the load of each server is large, the number of servers can be increased to solve the problem.

E. Basic Principle of System Implementation

Real-time monitoring software inference engine [10][12][13] to achieve the basic principles shown in Figure 2.

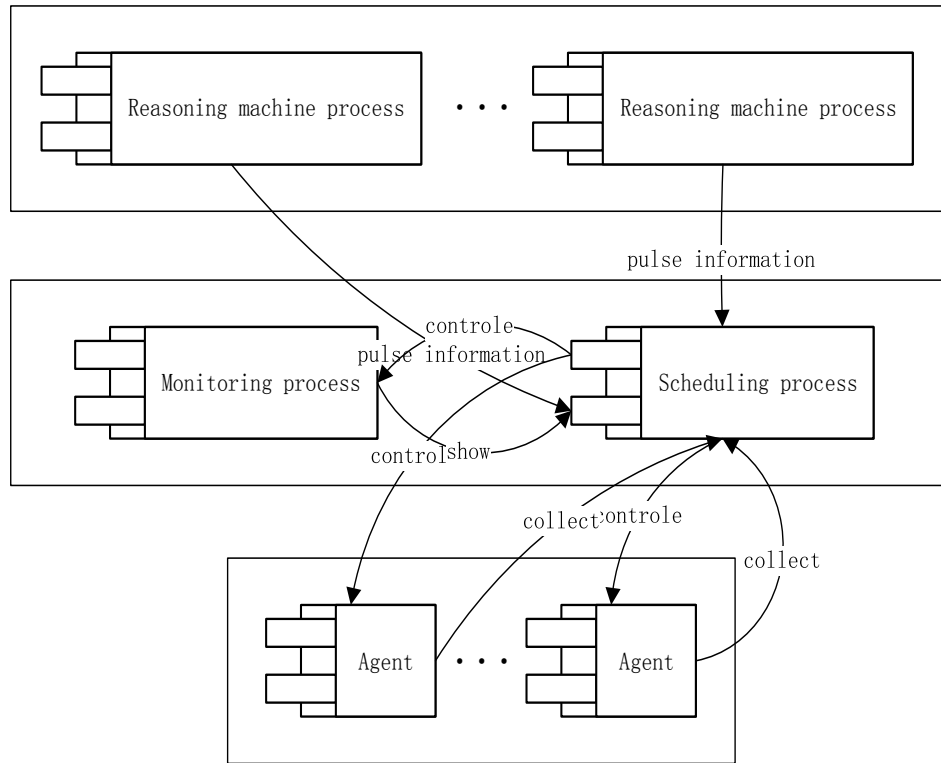


Figure 2. Basic Principle Diagram of System Implementation

The inference engine real-time monitoring software includes: system monitoring client process, process monitoring scheduling process, process monitoring service process. The basic principle of the system operation is as follows:

1) Pulse Information

Each monitored inference engine process sends its own pulse information. The pulse information is sent once every 1 second and broadcast by UDP communication. If the process monitoring process

can receive the pulse information within 5 seconds, the target process is running normally; otherwise, deal with the target process running abnormally.

2) *Process Monitoring Service Process*

The system monitoring service process starts automatically as the operating system starts. The system monitoring service process is responsible for starting the system scheduling service process, starting the process, stopping the process, downloading and installing the software package, obtaining the system running load information, obtaining the process running status information, obtaining the system running log information, monitoring the process running status, and broadcasting the process pulse information.

3) *System Monitoring Client*

System running status monitoring client mainly includes: management server running information, management process running information, management exception handling and TCP communication. The management of server operation information is mainly responsible for the display and maintenance of server operation parameters; managing process running information is responsible for displaying process running information, managing process, etc. Managing exception handling mainly deals with system running exceptions; TCP communication module is used to communicate with the process of process scheduling subsystem, send operation commands and receive operation results.

4) *Process Monitoring Scheduling Process*

The system monitoring service process starts with the operating system, and then starts the system scheduling service process according to the 'election' algorithm. The process monitoring scheduling process is mainly responsible for monitoring the running state of the target process, monitoring the server running information, maintaining system load balancing, and managing the target process. The following details:

a) *Monitoring the Running State of the Target Process*

The process monitoring and scheduling process receives the pulse information of the target process in real time, and continues if the pulse information

of the target process is received in timeout; otherwise, set the run state of the target process to a 'fail' state.

b) *Monitor Server Running Information*

The process monitoring scheduling process can send 'Get Server Average Load' or 'Get Server Real Time Load' commands to the process monitoring service process of the monitored server. Send server load information when the process monitoring scheduling module requires server load information.

c) *Maintenance of System Load Balancing*

Process monitoring scheduler selects the target process to migrate, the source server to migrate, and the target server to migrate according to the scheduling algorithm. See Figure 3 for specific scheduling algorithms and process migration processes.

d) *Management Target Process*

The management target process includes: starting the target process, stopping the target process, migrating the target process, downloading the installation server software package, upgrading the server software package, process master-slave switching, obtaining the system running load information, obtaining the process running state information, obtaining the system running log information, monitoring the process running state, broadcasting the process pulse information.

Process monitoring scheduling process realizes process scheduling according to scheduling algorithm and maintains system load balancing. Process scheduling has two ways: automatic scheduling and manual scheduling. The automatic scheduling algorithm [11] is shown in Figure 3.

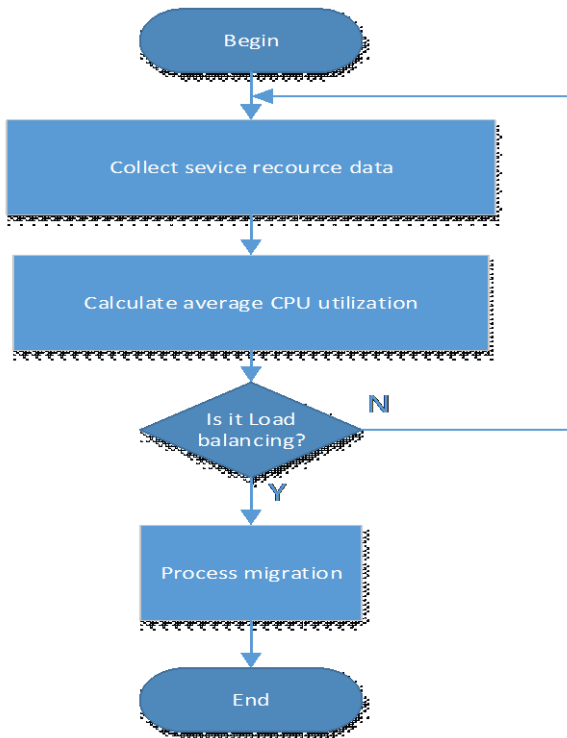


Figure 3. Process Scheduling Algorithm

The average CPU utilization formula is as follows:

$$\bar{U} = \frac{1}{N} \sum_{i=1}^N u_i \quad (1)$$

Where: \bar{U} represents average CPU utilization, u_i represents the i th CPU utilization, N represents the number of CPUs owned by the server.

Manual scheduling is the process and server of migration selected by spacecraft users. At this time, the migration mode of the migrated process is set as 'manual migration'. Then, the system scheduling service process migrates according to the above scheduling process.

III SOFTWARE ARCHITECTURE

In order to realize the running goal of all servers and service processes on the monitoring network, the system is divided into three parts: system running status monitoring client, system scheduling service process, system monitoring service process. System running status monitoring client running in one or more clients; the system scheduling service process runs on a certain server and is initiated by the system monitoring service process; the system monitor service process runs on each monitored server and starts with the operating system. The working process of inference engine real-time monitoring software is shown in Figure 4.

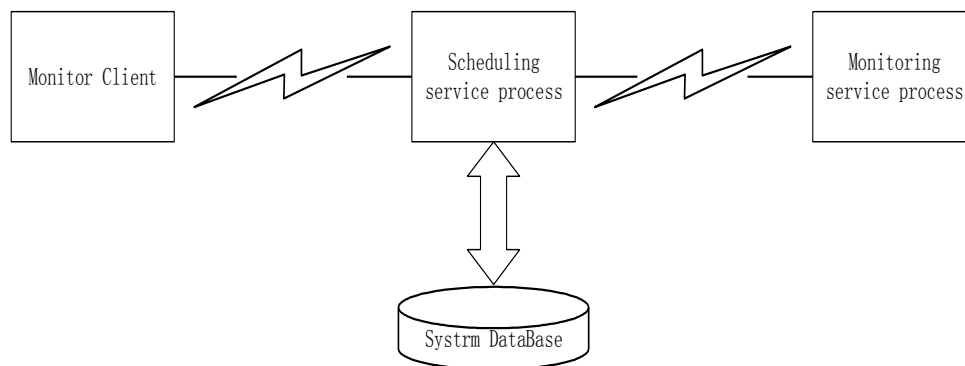


Figure 4. Software Architecture Diagram

The monitoring system status information consists of three parts: system operation status monitoring client, system scheduling service process and system monitoring service process.

A. System Running Status Monitoring Client

System running status monitoring client main modules: server management module, process management module, exception handling management module and TCP communication

module. The server management module is mainly responsible for the display and maintenance of server operating parameters; process management module is responsible for displaying process running information, managing process, etc. Exception processing module mainly deals with system operation exception; TCP communication module is used to communicate with the process of process scheduling subsystem, send operation commands and receive operation results.

1) Display System Status Information

System running state information mainly includes: server running information, service process running information, etc. TCP communication module receives service process operation information, and process management module is responsible for displaying service process operation information to spacecraft users; TCP communication module receives server operation information, and server management module is responsible for displaying server operation information to spacecraft users.

2) Display System Abnormal Alarm Information

TCP communication module receives system exception alarm information, exception handling module is responsible for alarm and display exception information to spacecraft users.

3) Handling System Operation Anomalies

When spacecraft users process system exception alarm information, if they choose 'save exception handling process', exception handling module sends exception handling process information in XML form to process scheduling subsystem through TCP communication module.

B. System Scheduling Service Process

System scheduling service process mainly consists of TCP communication server module, command manager, command processor, information manager, information processor, TCP communication client module, process scheduling management module and master-slave switching module. TCP communication server module is responsible for communication with all the system running status monitoring client; command

manager is used to save the received operation commands; command processor is responsible for reading and parsing commands; information manager is used to save system operation information; information processor is responsible for reading and parsing system operation information; TCP communication client module is responsible for communication with all process monitoring service modules; the process scheduling management module is responsible for starting and stopping processes.

1) Receiving Monitoring Client Operating Commands

The receiving and monitoring client operation commands are completed by independent threads. Monitors communication between client and system scheduling service processes via TCP. The monitoring client sends the monitoring client operating commands to the system scheduling service process through the TCP communication module. The TCP communication server module of the system scheduling service process is responsible for receiving the monitoring client operating commands. Monitor client operating commands received and written to the command manager.

2) Handling Monitoring Client Commands

The command processor is completed by independent threads. Command processor scan command manager. If there is an unprocessed command, read the command and parse it, and then select a TCP communication client to send the command to the system monitoring service process; if the command manager is empty, then wait for 1 millisecond to continue scanning.

3) Receiving System Monitoring Service Process Operation Information

Receiving system monitoring service process running information with independent threads to complete. After receiving the running information of the system monitoring service process, the TCP communication client module is written to the information manager.

4) Processing System Monitoring Service Process Operation Information

The information processor is completed by independent threads. Information processor scan information manager. If there is an unprocessed information, the information is read and analyzed, and then a TCP communication server is selected to send the information to the monitoring client; if the information manager is empty, then wait for 1 millisecond to continue scanning.

The system scheduling service process has two states: the main process state and the standby process state. When the main process runs, the standby process runs in an inhibitory manner; when the main process fails, the standby process switches to the main process state. When the scheduling process is in the host state, the process of switching the standby state is shown in Figure 5.

5) Process Management

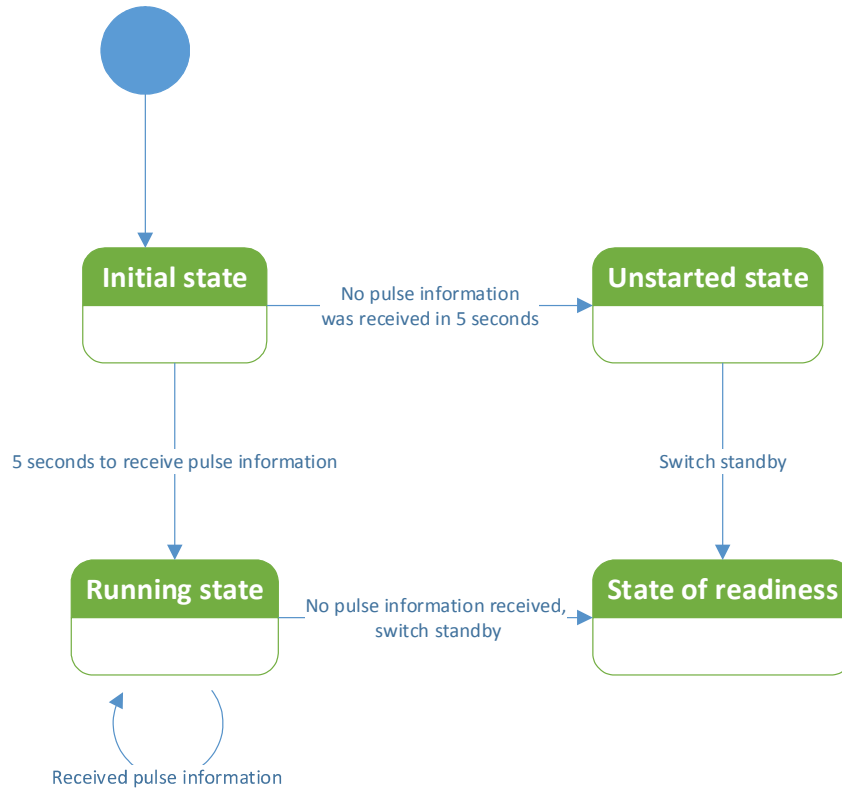


Figure 5. Host Switch to Standby Process

After the system monitoring service process starts, it enters the 'standby' state. If the host information is not received for 5 seconds, it enters the 'election' state; automatic transition to 'counting' status after 10 seconds from 'election' status; in the 'counting' state, if the 'minimum load', broadcast 'I am the host', 5 seconds into the 'host' state; if the 'load is large' then into the 'standby' state; if you receive 'I am the host' in 'Host' state, go to 'Standby' state.

When the scheduling process is standby, the process of switching host state is shown in Figure 6.

When in the 'initial state', if the scheduling module receives the pulse information to the 'running' state, if the scheduling module does not receive the startup information to the 'not running' state; when in the 'run' state, if the scheduling module does not receive the pulse information to the 'abnormal' state; when 'not started', the scheduling module selects the server to go to 'ready to start' state, if the scheduling module does not receive the start information go to 'clean' state;

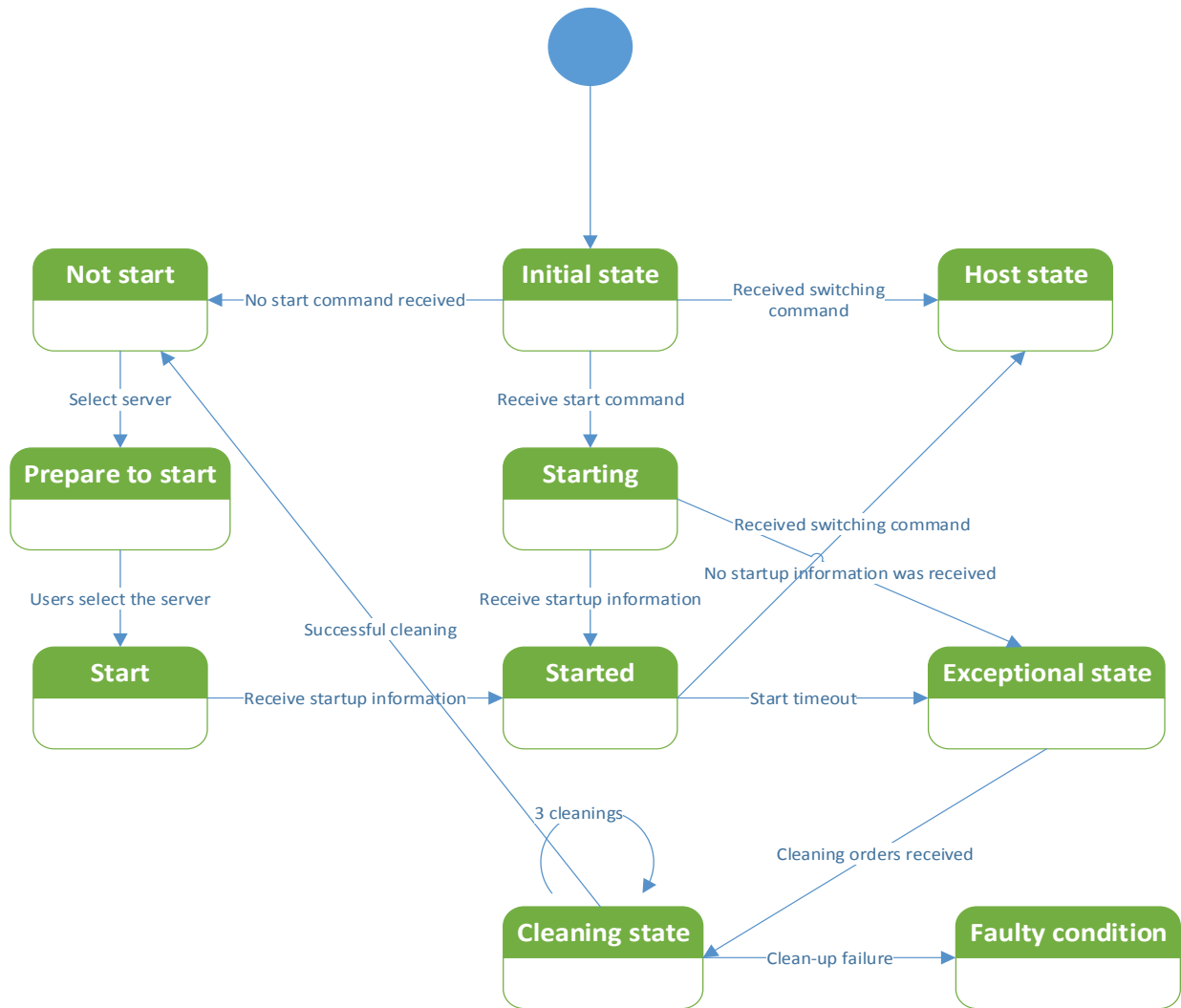


Figure 6. Standby Switch to Host Process

When 'ready to start', the scheduling module selects the server to go to the 'start to start' state. If the server is not installed, the scheduling module goes to the 'fault' state. When 'start', the scheduling module sends the start command to the 'start' state, and if the server is selected to have no software package installed, the scheduling module is transferred to the 'fault' state; when in the 'start', if the scheduling module receives the start information to the 'start' state, otherwise into the 'abnormal' state; when in the 'abnormal' state, the scheduling module directly turns to the 'clean' state; when in the 'clean up' state, if the clean up 3 times failed to go to the 'failure' state; when in the 'fail' state, when the user discovers a failure, manually set to the 'initial' state.

6) Send Process Running Status Information

When the monitoring client needs the process running state information, the command to obtain the process running state is sent to the system scheduling service process. The system scheduling service process command processor parses the command, and then obtains the process running state information according to the command parameters, and finally sends it to the monitoring client in XML form through the TCP communication server.

7) Pulse Information Of Broadcasting Process

When the system scheduling service process is running, the pulse information is broadcast every 1 second. If the system monitoring service process does not receive pulse information, the system monitoring service process restarts the system

scheduling service process according to the election algorithm.

C. System Monitoring Service Process

Handle system scheduling process commands. Command processor scan command manager. If there is an unprocessed command, read the command and parse it, then execute the command and send the execution results to the system scheduling service process through the TCP communication module; if the command manager is empty, then wait for 1 millisecond to continue scanning [14] [15].

The process monitoring subsystem is mainly composed of TCP communication module, command manager and command processor. TCP communication module is responsible for communication with process scheduling subsystem; command processor is used to save system maintenance client operation commands; command processor is used to process system maintenance client operating commands.

1) Receiving System Scheduling Service Process Commands

The operation command of the receiving process system scheduling reset process is completed by an independent thread. System monitoring service process and system scheduling service process communicate through TCP. The TCP communication module of the system monitoring service process is responsible for receiving the operation commands of the system dispatching service process. System scheduling service process operation commands are received and written to the command manager.

2) Handling System Scheduling Service Process Operation Command

The command processor is completed by independent threads. Command processor scan command manager. If there is an unprocessed command, read the command and parse it, then execute the command and send the execution results to the system scheduling service process through the TCP communication module; if the command manager is empty, then wait for 1 millisecond to continue scanning.

3) Start the System Scheduling Service Process

The system monitoring service process starts with the operating system, and then starts the system scheduling service process according to the 'election' algorithm.

4) Pulse Information of Broadcasting Process

According to the design requirements, the target process broadcasts pulse information outward UDP every second. The definition of pulse information includes: host unique identification, host IP, TCP port, process identification, running identification and other information.

IV CONCLUSION

Multi-inference real-time monitoring software has the advantages of real-time monitoring, automatic scheduling and load balancing, which can make the monitored inference process run stably and reliably for a long time without anybody on duty. The author believes that the system can not only monitor the running state of multiple inference engines in real time, but also be widely used in many occasions that require process monitoring. At present, there is no domestic application in this area, it can be said that the product to fill this gap.

ACKNOWLEDGMENT

Fund Project: Supported by the Special Project of Shaanxi Provincial Department of Education (No.: 17JK0388).

REFERENCES

- [1] Gaoli, Bu Huaiyuan, Hushu. A medical diagnosis reasoning machine design and implementation [J]. Computer application and software, 2002 (6): 44-46.
- [2] Gu Shenming, Liu QuanLiang. An expert system is based on Web design and implementation [J]. Computer engineering. 2001, 27 (11): 100-101.
- [3] ZhaoZidu. Reasoning mechanism and reasoning method [J]. Automation expo, 1997 (6): 19-20.
- [4] Wu Quanyun, etc. Artificial intelligence and expert system [M]. Anhui: national defense science and technology university press, 1995, 168-172.
- [5] ChenZhaoQian, et al. DBEST: apractical fault diagnosis expert System development tools [J]. Microcomputer, 1995, 15 (6): 16-20.

- [6] Guo Huawei, Shi WenKang, DengYong, et al. Conflict of evidence:discard, discover or to solve [J]. Systems Engineering and electronics, 2007, 29 (6): 890-898.
- [7] PaksoyA, GktürkM. Information fusion with Dempster-Shafer evidence theory for software defect prediction [J]. Procedia Computer Science, 2011 (3): 600-605.
- [8] Hartley RVL. Transmission of information [J]. Bell Systems Technical Journal, 1928, 7 (3): 535-563.
- [9] JousselmeAL, GrenierD, Bosse. A new distance between Two bodies of evidence [J]. Information Fusion, 2001, 2 (2): 91 – 101.
- [10] 5th National Conference on Artificial Intelligence, Philadelphia, USA, Aug. 11 – 15: 869-901.
- [11] Honsel Daniel, Herbold Verena, Waack Stephan et al. Investigation and prediction of open source software evolution using automated parameter mining for agent-based simulation [J] Automated Software Engineering, 2021, 28(1).
- [12] Yinling Liu, Tao Wang, Haiqing Zhang et al. An improved approach on the model checking for an agent-based simulation system [J] Software and Systems Modeling, 2020.
- [13] Chouaki Tarek, Puchinger Jakob Agent based simulation for the design of a mobility service in the Paris-Saclay area [J] Transportation Research Procedia, 2021, 52.
- [14] Feather Christopher Footing the Reconstruction Bill: An Appraisal of the Financial Architecture for Disaster Rebuilding in the United States of America [J] International Journal of Disaster Risk Reduction, 2021.
- [15] Bi Tingting, Ding Wei, Liang Peng et al. Architecture information communication in two OSS projects: The why, who, when, and what [J] The Journal of Systems & Software, 2021, 181.