


Research Article

Implementation of Dynamic Load Balancing in Distributed System Based on Improved Algorithm

Guangyu Zhou 

College of Digital Technology and Engineering, Ningbo University of Finance and Economics, Ningbo, Zhejiang, China

Correspondence should be addressed to Guangyu Zhou; zhouguangyu@nbufe.edu.cn

Received 15 April 2022; Revised 7 June 2022; Accepted 24 June 2022; Published 6 July 2022

Academic Editor: Imran Shafique Ansari

Copyright © 2022 Guangyu Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The digitalization and networking of the operating status of manufacturing equipment and facilities are still carried out on the basis of the automatic measurement system. Combining the embedded microprocessor AD/DA device and the network controller into an IC chip not only can solve the technical problems between the embedded microcontroller and the Internet but also can reduce the cost of this connection to a minimum limit. Embedded intelligent technology, especially the development and application of agent technology, makes the monitoring system change from a central computing model to a distributed model. With the application of computing from centralized development to distributed development. CBA has also transitioned from static central control to dynamic distributed control. The system load balancing method, distributed in the system processor, can enhance the capabilities of all the instruments and equipment in the system; eliminate the imbalance between busy and idle, and improve the overall performance of the system. The purpose of this article is to analyze and experimentally verify the parallel computing methods of computer distributed systems, and it can be applied to parallel computing for a variety of environmental support in computer systems. An effective combination of distributed object technology and embedded system technology can build a model of a distributed parallel computing system. After the distributed parallel computing system retains the advantages of the previous system, the distributed availability of parallel computing systems has been greatly improved.

1. Introduction

Up to now, the digitization and networking of the operating status of manufacturing equipment and facilities are still carried out on the basis of the automatic measurement system. This system is composed of industrial PCs, data acquisition equipment, and network equipment controllers and is limited by the structure of the equipment. Installation space and system cost also have a certain degree of restriction. Embedded system engineering as a new way and method can successfully solve this problem. The progress of embedded system engineering has proposed a new way to solve this problem. Silicon microelectronics technology can make microelectromechanical systems more mature in a single IC chip. Combining the embedded microprocessor AD/DA device and the network controller into an IC chip can not only solve the technical problems between the embedded microcontroller and the Internet but also reduce the cost of this connection to a minimum limit. Embedded

intelligent technology, especially the development and application of agent technology, makes the monitoring system change from a central computing model to a distributed model.

In the real society, information technology represented by the Internet, embedded system, and virtual reality technology are developing in the direction of digitalization, intellectualization, and networking. It has been widely used in manufacturing. By seamlessly connecting information technology and intelligent systems, computing, a technology with relatively high performance, becomes the core technology of computing systems. At the same time, it is also the main research direction and research goal in the computer research industry. It has been highly praised by the computer industry of many countries in the world and has important research value and important role. Many types of computers have been developed for many years. High-performance computing has become a major indicator of a country's technological level, and it is a very important part of the

computer science system. The introduction of the Galaxy series and high-performance computers also proves that our country has excellent technology in high-performance processing and has made significant progress in scientific research.

2. Related Work

Document records that, with the application of computing from centralized development to distributed development, CBA has also transitioned from static central control to dynamic distributed control [1]. The document records that the distribution of control tasks, the autonomy of the control subject, and the coordination between them are getting closer [2]. The document records that early CBM systems usually use static centralized control structures [3]. The document records that most of the current CBM systems adopt a hierarchical control structure [4]. This model uses modularity and hierarchical control behavior to divide the control function into multilevel control units when designing. The document records that in the hierarchical control structure; each level will lower the task to the next level [5]. After the control instruction is decomposed, the actuator completes the operation specified by the system. The document records that sensors use the same method for feedback, but the direction is opposite, and the direction of control information flow is opposite [6]. The advantage of the hierarchical control structure is to enable the system to develop modularly and establish a standardized model. However, the document records that the rigid relationship between the upper and lower layers will lead to the transmission and feedback of information, which is executed in a strictly hierarchical manner, so the responsiveness and scalability of the system can be reduced [7]. Therefore, the document records that in the distributed control mode, the control structure of the high-speed communication network will form a distributed computing economic system [8]. The document records that the roughly independent units can be differentiated according to the application system, and their master-slave relationship has autonomy for these units [9]. They have a parallel relationship with each other and carry out information conversion and collaboration through a preset communication interface. The document records that smart sensors are used in a large number of distributed CBM systems [10]. The document records that the organization and management of sensor networks and their communication and collaboration will face huge difficulties and these resources are distributed in a certain area [11]. Because of this situation, the document records that the existing PC method cannot be used to model the sensor network management with the organization structure scheme [12]. People will make the agent and holon structure have many advantages through this method [13]. It can simplify the sensor modeling procedure and provide a new path for sensor communication and cooperation. Load balancing refers to the transfer of tasks from heavy processors to light processors.

The system load balancing method, distributed in the system processor, can enhance the capabilities of all the

instruments and equipment in the system and eliminate the imbalance between busy and idle so that the overall performance of the system is improved [14]. Load balancing research includes static and dynamic situations [15]. Due to the increasing maturity and improvement of the technology faced, distributed technology is applied to computing systems in all walks of life, such as distributed computing, artificial intelligence, and so on [16]. The object of distributed computing is distributed technology. The application area is to enable the technology to be fully and fully implemented in a heterogeneous network environment so that the complexity of the control system in the process of development, management, and maintenance is limited [17]. The use of technology-oriented advice is used to construct a system from the real world, and in the construction of the system, people's natural thought patterns are maximized [18]. At the same time, people's logical thinking can also be applied to the system. The object-oriented construction of the system makes the system have the same understanding of the system with objective events [19]. The system interface is cleared to facilitate the management of the system, and the scalability of the system will not be affected by changes in details, and it can be reused and maintained. Traditional theoretical methods are used in distributed object technology. However, due to the proliferation of calculation methods, many traditional object thoughts cannot be tried [20, 21]. Put forward various challenges to the traditional object principle to improve and develop the distributed computing system.

3. Research on Distributed Matrix Multiplication Parallel Computing System

3.1. System Research. The performance of a parallel computer system mainly includes three aspects: processing unit, network topology, and parallel computer algorithm. The processing unit is the main part of the parallel computer system. The high-performance processing unit can improve the system performance, reduce the volume and loss of the system, reduce the complexity of the system, and make the software maintainable. He is mainly responsible for the communication commands between the main processor and the main processor and the interaction between the control logic unit and the main processor. The PE array mainly has three registers to complete. Reg0 is a parameter register; REG1 is a command register; and REG2 is a status register. The specific situation is shown in Table 1.

From the perspective of the development of neutral connection mode, the connection network is an improvement of parallel single segment sharing and pcvpci-x. It is an improvement from parallel single segment sharing and pcvpci-x. In It is in VEM midline to point-to-point mode. For example, Ethernet is used for bus packet switching, and switches are used to convert the topology structure, so as to improve the bandwidth and scalability of VO connection, so that VO shared bus can overcome the original shortcomings and support access to multiple systems, the functionality has been greatly improved. The details are shown in Table 2.

TABLE 1: Control logic unit and PE array command interaction register content.

Position	Name	Reset value	Description
REG0	Parameter register		
0~12	—	0	Save
13~15	A_Row	0	Enter the value of matrix A
16~20	—	0	Save
21~23	B_Column	0	Enter the value of matrix B
24~31	—	0	Save
32~47	A_Column	0	Enter the value of matrix A
48~62	—	0	Save
63	More	0	Matrix multiplication calculation
REG1	Command register		
0	Start	0	Calculation start
1~61	—	0	Save
62	Rst_n	0	PE reset
63	Terminate	0	Terminate matrix multiplication calculation
REG2	Status register		
0~7	Version	—	Matrix multiplier
8~62	—	0	Save
63	Status	1	Matrix multiplier calculation

TABLE 2: Comparison of main characteristics of PCI-e, Ethernet, and serial Rapid I/O.

	PCI-e	1000BASE-T Ethernet	Rapid I/O
Topology	Tree-like	Arbitrary	Arbitrary
Multicast aspect	No	Yes	Yes
Send and receive signals	1x, 2x, 4x, 8x, 12x, 16x, 32x	4x	1x, 4x
Transmission distance	~50–60 cm + 3 connectors	100 m	~90–110 cm + 3 connectors
Data	2.7 GBaud per channel	2000 Mbps	1.26, 2.6, 3.126 G per channel
Clock	Embed	Embed	Embed
Time delay	High	Highest	Low
System protocol stack	High	Highest	Low
Software and hardware situation	Rich	Most abundant	Richer

3.2. Performance Analysis of Parallel Computing System.

The time of the parallel algorithm depends on the size of input data and the number of processing units, as well as the speed of computation and process communication. Therefore, the parallel algorithm cannot complete the accurate evaluation of the parallel system. Parallel runtime is the time from the beginning of parallel to the completion of the processor. T_S represents the serial running time, and T_P represents the parallel running time.

The cost function of a parallel system is the time consumed by all processors minus the time consumed by a single processor. When solving the problem, the processor's

total time is PTP ; T_S is the time consumed to complete the work; and the rest is spent. If T_o is used to represent the cost function of the system, the total cost of the system can be expressed as follows:

$$T_O = P \cdot T_P - T_S. \quad (1)$$

When solving the problem, it is measured by the speedup ratio. Speedup refers to the comparison between the time consumed by a single processor and the time consumed by the entire processor. Using S to represent the speedup ratio, it can be expressed as follows:

$$S = \frac{T_S}{T_P}. \quad (2)$$

Efficiency is to measure the working time in the processor. It is the ratio between the number of processors and the speedup. Using E to represent efficiency, then efficiency E can be expressed as follows:

$$E = \frac{S}{P}. \quad (3)$$

Redundancy can be expressed as follows:

$$R = \frac{O_P}{O_S}. \quad (4)$$

The parallel speedup can be expressed as follows:

$$S = \frac{T_{\text{serial}}}{T_{\text{parallel}}} = \frac{T_S + T_{P(1)}}{T_S + T_{P(1)}/P}. \quad (5)$$

There is a certain connection between subtasks, and the processor will adjust the data communication and idle state

$$T_k = T_{\text{comp},k} + T_{\text{comm},k} + T_{\text{sync},k} + T_{\text{idle},k} \quad (1 \leq k \leq P). \quad (6)$$

The completion time of all subtasks is the working time of the parallel system

$$T_P = \max(T_k) \quad (1 \leq k \leq P). \quad (7)$$

If the algorithm redundancy $r = 1$, it can be expressed as follows:

$$T_s = \sum_{k=1}^P T_{\text{comp},k}. \quad (8)$$

In order to enhance the concurrency of the system, if the algorithm is changed and the serial program is added to the subtask, the redundancy will be enhanced

$$T_s < \sum_{k=1}^P T_{\text{comp},k}. \quad (9)$$

At the same time, the redundancy of parallel algorithm can be expressed as follows:

$$R = \frac{\sum_{k=1}^P T_{\text{comp},k}}{T_s}. \quad (10)$$

Gustafson's law can be expressed as follows:

$$S' = \frac{T'_{\text{serial}}}{T'_{\text{parallel}}} = \frac{T_S + PT_{P(1)}}{T_S + T_{P(1)}}. \quad (11)$$

3.3. Performance Analysis of Matrix Multiplication Parallel Computing System. The system uses the multicast method to calculate and send data. If the time of data communication time assessment processing unit does not cross, whether the system uses one- or two-dimensional method, it can be expressed as follows:

$$T_p = T_{\text{comm}} + T_{\text{comp}} = 2N^2t_w + \frac{N^3}{P}. \quad (12)$$

The speedup ratio and efficiency of the system are calculated as follows:

$$S = \frac{T_S}{T_p} = \frac{N^3}{2N^2t_w + N^3/P} = \frac{1}{2t_w/N + 1/P}, \quad (13)$$

$$E = \frac{S}{P} = \frac{1}{1 + 2Pt_w/N} = \frac{1}{1 + OP/N}.$$

When two matrices are multiplied, the time of the first matrix algorithm is basically the same as that of a single matrix. But, in the second matrix algorithm, if the two-dimensional method is used, the two-dimensional transmission time will increase

$$T_{\text{comm}2} = 2N^2t_w + 2N^2t_w = 4N^2t_w,$$

$$T_{p2} = T_{\text{comm}2} + T_{\text{comm}2} = 4N^2t_w + \frac{2N^3}{P}, \quad (14)$$

$$S_2 = \frac{T_S}{T_{p2}} = \frac{2N^3}{4N^2t_w + 2N^3/P} = \frac{1}{2t_w/N + 1/P},$$

$$E_2 = \frac{S_2}{P} = \frac{1}{1 + 2Pt_w/N} = \frac{1}{1 + OP/N}.$$

In a one-dimensional partition, because the calculation time of the second matrix is exactly the same as that of the first matrix, the communication cost of the system is the same as that of the single calculation

$$T_{p1} = T_{\text{comm}1} + T_{\text{comm}1} = 2N^2t_w + \frac{2N^3}{P},$$

$$S_1 = \frac{T_S}{T_{p1}} = \frac{2N^3}{2N^2t_w + 2N^3/P} = \frac{1}{t_w/N + 1/P}, \quad (15)$$

$$E_1 = \frac{S_1}{P} = \frac{1}{1 + Pt_w/N} = \frac{1}{1 + OP/N}.$$

The multiplication of two matrices is extended to K matrices. When a two-dimensional partition is used, every matrix multiplication has two transmission processes

$$T_{\text{comm}2} = 2N^2t_w + 2(k-1)N^2t_w = 2kN^2t_w,$$

$$T_{p2} = T_{\text{comp}2} + T_{\text{comm}2} = 2kN^2t_w + \frac{kN^3}{P},$$

$$S_2 = \frac{T_S}{T_{p2}} = \frac{kN^3}{2N^2t_w + kN^3/P} = \frac{1}{2t_w/N + 1/P}, \quad (16)$$

$$E_2 = \frac{S_2}{P} = \frac{1}{1 + 2Pt_w/N} = \frac{1}{1 + OP/N}.$$

One-dimensional partition is studied. If only the transmission time of the first matrix algorithm does not cross the time of other algorithms, the communication cost of multiple matrix computing systems can be expressed as follows:

$$T_{\text{comm}1} = 2N^2t_w. \quad (17)$$

4. Design of Distributed Network Parallel System Based on Embedded System

4.1. System Model Design. In general, distributed parallel computer system consists of several nodes communicating with each other through a message transmission system. The details are shown in Figure 1. There are two types of processes on each node machine: one is user-oriented application process, and each node machine can have an application layer composed of many application processes. The other is the scheduling process; each node has only one scheduling process, which is suitable for the system control layer, and the scheduling process of each node has a higher priority than the application process. It contains a set of uncoded tasks, some of which can be attributed to the application process of the local node or sent to the adjacent node. Each scheduling process communicates with the scheduling process of adjacent nodes to send or receive the task or result of the programming process on its own node.

There are four different message types in the scheduling system: task message, result message, loadable mode, and standby message. In Figure 1, it is represented by send (*, *, *, *). The first two parameters are I and messages. They are the source process and the target process of messaging. It can be a scheduling process or an application process. The third and fourth parameters are message content and message type, respectively. Wrđ and I are used to represent message task result load message and waiting message, respectively, in the figure. At present, the commonly used distribution algorithms mainly include the probability of randomly selecting task nodes for scheduling. According to the load difference and additional contract algorithm, no matter which scheduling algorithm is used, the following three problems must be solved: (1) when to start the load period, (2) what are the source and target nodes of each balanced scheduling, and (3) which task should be selected for scheduling. In addition, according to the classification of local liabilities, initiators can be divided into four types: sender, receiver and driver and instruction.

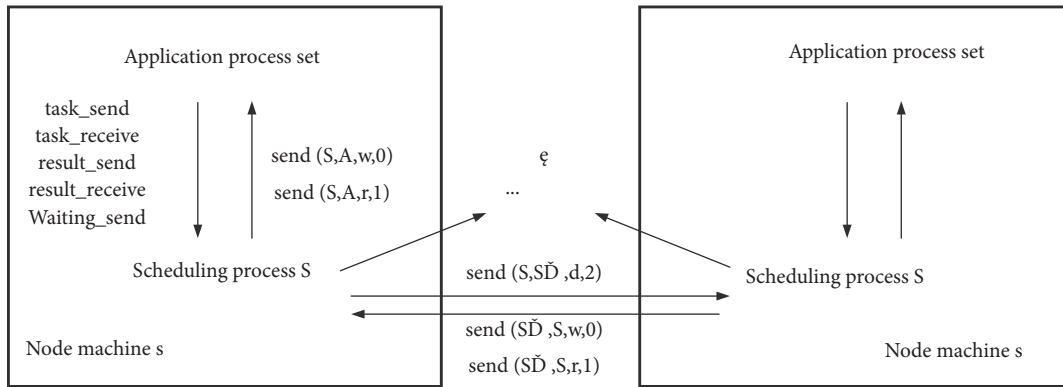


FIGURE 1: Load balancing model.

4.2. *Structure Design of Embedded System.* Sensors are the front-line components of a measurement system. It is the input variable into easy-to-measure communication. According to the relevant national standards, the concept of the sensor in the standard is to feel the measured signal and convert it into an output signal according to some rules. In general, the sensor is the equipment that we think can convert the value of the SARS signal into measurable. Literally, the above two statements mean more than that. No matter how defined by IC or national standards, the type of input variable is not limited to the value of nonelectrical signal. In fact, in the field of measurement and control, there are also embedded industrial device gateways, even virtual sensors, and other embedded industrial device gateways. As a network protocol conversion equipment, to connect the Tesco communication equipment to the corresponding measurement and control system, it needs to install the appropriate sensor channel and output data according to the set model. Provide some variable embedded device gateway logic structure that cannot be measured directly, as shown in Figure 2.

The information processing unit is mainly composed of several independent submodules, such as communication interface protocol interpreter, policy manager and service interface, and submodules of the shared control unit. It includes the communication interface module. The emagent is responsible for maintaining and updating the group's policy pool. The service interface module provides the maintenance and service of the service pool, as well as the mapping of the screen and the security of access. Using the planned operation mode, the service access only needs to formulate the service code and provide the parameters needed to realize the service. This method not only reduces the length and text of communicators, so it always processes data at the data source. It also ensures that the system is always running under automatic control. The first function of a task manager is task scheduling, which transforms the received emergency agent request into a task activity code and is responsible for repairing and replacing the task pool. The second task is task execution, which is responsible for task scheduling and executing active tasks. A task pool is a two-level priority task stack, which has the function of cache, and at the same time, urgent tasks need to be put first. The

production of remajent is the data value with specific meaning.

According to the data model, the data processor not only physically processes the basic data stream sent. It is a technical quantity, and the technical quantity sent to the physical part is also transferred into a recognizable data format. The physical processing unit consists of four modules: device planning, service interface, description table interface, and device driver. At the same time, there is a shared controller interface as a submodule. The information module in the physical channel is responsible for data processing. This module is the main part of the physical system. All tasks are carried out under the control of the equipment planning module. The device description module in the interface table is responsible for the update and maintenance description of the device and provides the corresponding access rights in the application program interface. The driven program is responsible for providing an access system to the device channel.

4.3. *Design of Hybrid Load Balancing Algorithm.* Agent based on the computer is a new method to realize distributed computing. In recent years, with the development of distributed artificial intelligence, agent technology has become more and more mature. If we can make use of the technology of the agent to execute dynamically, we can adjust the network load by scheduling the agent as the sender, which can greatly improve the situation of distributed parallel computing. The system to realize the agent is the event processing system.

4.4. *Analysis of Test Results.* In numerical parallel computing, the CPU of computer nodes is fully utilized, so it is impossible to describe a load of each node by CPU utilization. In view of this, we believe that: after task allocation, each data node performs the calculation according to the actual amount of the specific algorithm and its own performance index as the load factor. According to the meaning of load rate, Figure 3 shows the comparison of the FIFO algorithm and the MBL algorithm for task scheduling.

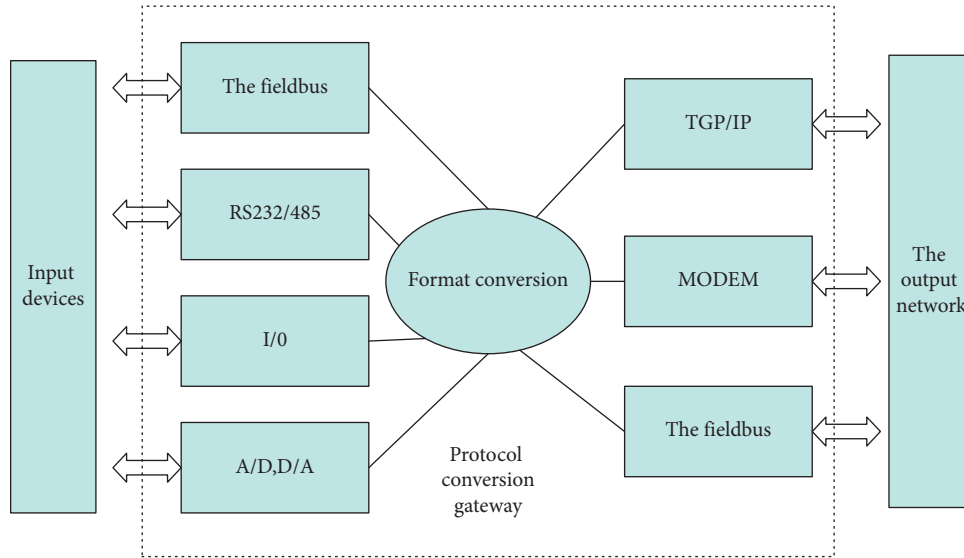


FIGURE 2: Schematic diagram of embedded protocol conversion network structure for field devices.

Figures 4 and 5, respectively, show the comparison of the load ratios obtained by using the FIFO algorithm and the MBL method. The load allocation is used to balance the task allocation of scheduling and distributed parallel simulation of the above parallel segment. As can be seen in Figure 4, when using the FIFO algorithm, the system imbalance is 14.69 degrees, and the maximum difference between the calculated node load ratio and 1 is 0.09 degrees; when using the MBL algorithm, the node imbalance is 3.11 degrees; and the maximum difference between the calculated node load ratio and 1 is 1.09 degrees. As can be seen in Figure 5, when using the FIFO algorithm, the system imbalance is 43.08 degrees, and the maximum difference between the calculated node load ratio and 1 is 0.42 degrees; when using the MBL algorithm, the node imbalance is 6.88; and the maximum difference between the calculated node load ratio and 1 is 0.9. Therefore, it can be concluded that the effect of the MBL algorithm is better than the FIFO algorithm.

In addition, it should be noted that the fourth step of the MBL algorithm is dynamic load balancing, which makes the MLB algorithm support fault tolerance and scalability of the distributed system very well. Now we can explain it in detail.

In some studies, the amount of network data in sensor networks shows a doubling trend with the increase in the number of sensors, and even data storm occurs. Data storm greatly increases the possibility of packet crash, which greatly reduces network communication. Ethernet is located in multiple subnets, each of which is a subconflict area, and the subconflict area is isolated by the switch. For distributed sensor networks, a large part of information exchange will be performed between sensor nodes; if a switch is used to separate the “partner” sensor group from other sensors in a relatively small area, the possibility of collision can be reduced. The influence of the number of nodes on the network communication delay is shown in Figure 6. With the increase of network node value n , the packet loss rate caused by P-Sensor end delay and collision

also increases correspondingly, indicating that the probability of packet collision will gradually increase with the increase of the number. When the number of sensor nodes in the subnet is less than 33, the networks delay changes little with the increase in the number of nodes. However, when the number of sensor nodes exceeds 65, the network delay and packet loss rate will increase rapidly. Especially when the number of sensor nodes exceeds 129, the network delay increases significantly, which indicates that the Ethernet is in a state of congestion. In practice, the number of sensor nodes can be determined according to the requirements of the network delay system, and the sensor network can be divided accordingly. If possible, the number of sensors in the subnet should be limited to 33 and not more than 65 as far as possible.

Another effective measure of Zhou’s network load is the data transmission frequency of Zhou’s network nodes. Figure 6 shows the end-to-end distribution of IP sensors and the trend that changes with the increase of message transmission frequency in different data transmission cycles calculated by RTT. Take LAN as standard, Ethernet; the number of network nodes is 33; and the length of the data packet is 1,025 nodes.

Another effective measure of Zhou’s network load is the data transmission frequency of Zhou’s network nodes. Figure 7 shows that in different data transmission cycles based on RTT calculation, IP sensors have been distributed end to end, and the change trend has increased. The Ethernet node is 33 bytes, and the packet length is 1,025 nodes.

As can be seen from Figure 7, in the system, when the data transmission frequency f requested by all nodes is relatively small, the end-to-end delay variation of the simulated data packet is not obvious, and the delay on the network is relatively stable, almost consistent with the number of communication days. The data packet is delayed, and the packet loss percentage is approximately the same. A

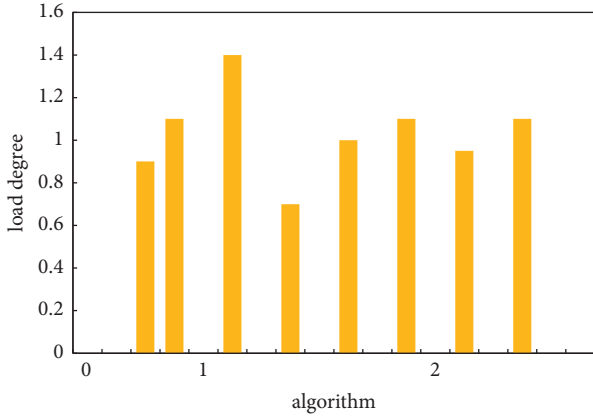


FIGURE 3: Load comparison between FIFO algorithm and MLB algorithm.

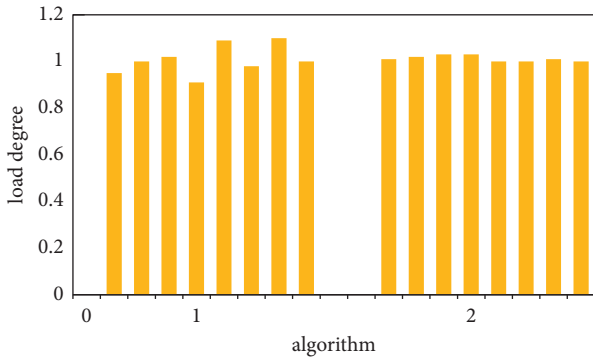


FIGURE 4: Load comparison between FIFO algorithm and MLB algorithm.

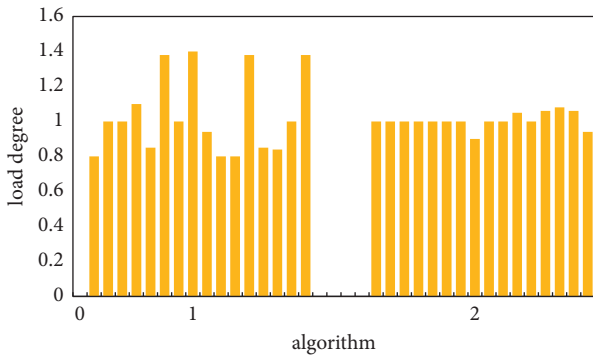


FIGURE 5: Load comparison between FIFO algorithm and MLB algorithm.

fixed low value means that the Ethernet bandwidth can meet the transmission of all information, and the probability of conflict with the data packet f is very small; even if there is a conflict, there will be almost no delay. When f rises to a certain value, the delay of the packet network will increase rapidly, and even the data transmission will fail. And the proportion of packet loss rate will rise sharply, which shows that in the case of network overload, the time delay is also a distinct feature of whether the network load is overloaded.

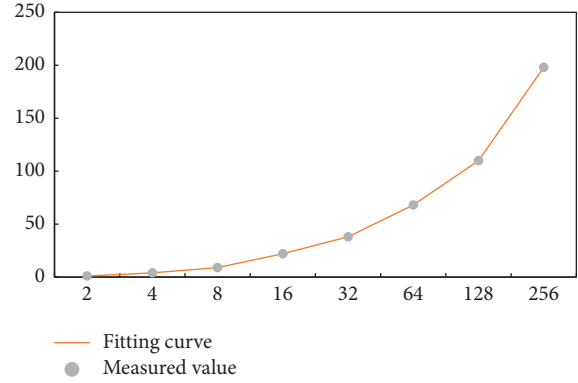


FIGURE 6: Delay characteristics of IP sensor networks with different number of nodes.

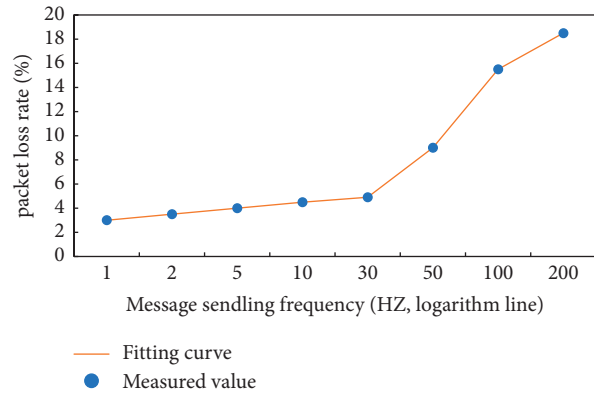


FIGURE 7: Delay characteristics of IP sensor networks with different packet transmission frequencies.

5. Conclusion

The purpose of this paper is to analyze and verify the parallel computing method of the computer distributed system. It can be applied to parallel computing for various environment supports in the computer system. By combining distributed object technology with embedded system technology, we can build a model of the distributed parallel computing system. After retaining the advantages of the previous system, the availability of distributed parallel computing system has been greatly improved. In addition, this kind of distributed parallel computing system is very important to the development of distributed parallel computing system; the reliability and convenience of the computer resource system are also greatly improved and developed.

Data Availability

The data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

- [1] L. X. Luo, "Information fusion for wireless sensor network based on mass deep auto-encoder learning and adaptive weighted D-S evidence synthesis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 11, pp. 519–526, 2018.
- [2] N. Maskey, S. Horsmanheimo, and L. Tuomimäki, "Latency analysis of LTE network for M2M applications," in *Proceedings of the 13th international conference on telecommunication*, pp. 1–7, Graz, Austria, July 2015.
- [3] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. S. Kweon, "High-quality depth map upsampling and completion for rgb-d cameras," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5559–5572, 2014.
- [4] A. F. M. Shahen Shah, H. Ilhan, and U. Tureli, "RECV-MAC: a novel reliable and efficient cooperative MAC protocol for VANETs," *IET Communications*, vol. 13, no. 16, pp. 2541–2549, 2019.
- [5] N. Ta, G. Li, Y. Xie, C. Li, S. Hao, and J. Feng, "Signature-based trajectory similarity join," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 870–883, 2017.
- [6] C. C. Huang, J. H. Li, C. L. Mei, and W. Z. Wu, "Three-way concept learning based on cognitive operators: an information fusion viewpoint," *International Journal of Approximate Reasoning*, vol. 83, pp. 218–242, 2017.
- [7] K. Do, T. Tran, and S. Venkatesh, "Graph transformation policy network for chemical reaction prediction," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 750–760, Anchorage AK USA, August 2019.
- [8] A. W. Astin and J. L. Holland, "The environmental assessment technique: a way to measure college environments," *Journal of Educational Psychology*, vol. 52, no. 6, pp. 308–316, 1961.
- [9] B. C. Oliveira, A. A. Seibert, V. K. Borges, A. Albertazzi, and R. H. Schmitt, "Employing a U-net convolutional neural network for segmenting impact damages in optical lock-in thermography images of CFRP plates," *Nondestructive Testing And Evaluation*, vol. 36, no. 4, pp. 440–458, 2021.
- [10] D. Sun, A. Li, B. Tang, and M. Wang, "Integrating genomic data and pathological images to effectively predict breast cancer clinical outcome," *Computer Methods and Programs in Biomedicine*, vol. 161, pp. 45–53, 2018.
- [11] L. Schneider, M. Jasch, B. Fröhlich et al., "Multimodal neural networks: rgb-d for semantic segmentation and object detection," in *Proceedings of the Scandinavian Conference on Image Analysis*, pp. 98–109, Tromsø, Norway, June 2017.
- [12] M. P. P. Kumar, B. Poornima, H. S. Nagendraswamy, and C. Manjunath, "A comprehensive survey on non-photorealistic rendering and benchmark developments for image abstraction and stylization," *Iran Journal of Computer Science*, vol. 2, no. 3, pp. 131–165, 2019.
- [13] W. Wang, E. Xie, X. Li et al., "Pyramid vision transformer: a versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, Montreal, QC, Canada, October 2021.
- [14] L. Du, "How much deep learning does neural style transfer really need? An ablation study," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 3139–3148, Snowmass, CO, USA, March 2020.
- [15] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale convattentional image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9981–9990, Montreal, QC, Canada, October 2021.
- [16] B. Buckholtz, I. Ragai, and L. H. Wang, "Cloud manufacturing: current trends and future implementations," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, Article ID 044001, 2015.
- [17] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," *Artificial Intelligence Review*, vol. 54, pp. 593–637, 2021.
- [18] S. Antol, A. Agrawal, J. Lu et al., "Vqa: visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, Santiago, Chile, December 2015.
- [19] I. Gatty, "Description of alternative measures of the concepts of vocational interest: crystallization, congruence, and coherence," *Journal of Vocational Behavior*, vol. 27, no. 1, pp. 37–55, 1985.
- [20] A. P. Singh and G. J. Gordon, "Matching users and items across domains to improve the recommendation quality," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 801–810, New York, NY, USA, August 2014.
- [21] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Madrid, Spain, October 2018.