Scholar Publishing Group
International Journal of Multimedia Computing
https://doi.org/10.38007/IJMC.2025.060115
ISSN 2789-7168 Vol. 6, Issue 1:152-159

Research on Load Balancing Technology in Distributed System Architecture

Buqin Wang

Meta Platforms / Infrastructure, Menlo Park, CA, 94025, US

Keywords: Load balancing; Distributed system; Adaptive algorithm; Machine learning; Blockchain

Abstract: Load balancing is crucial in distributed systems, as it helps optimize the rational allocation of resources and improve overall system performance and reliability. But as the system scale continues to expand, load balancing technology also faces complexity issues. The load balancing algorithm increases computational overhead in large-scale systems, and the dynamic changes in load are highly unpredictable, making it difficult for the system to adjust load allocation in real time. In addition, system security also faces severe challenges such as malicious attacks and information leaks. To address these issues, optimization strategies include adaptive load balancing algorithms that can automatically adjust loads based on system status. Apply machine learning and deep learning models to predict load changes and improve scheduling accuracy. Combining blockchain technology to enhance the security of the load balancing process, avoiding malicious attacks and data leaks. Through these strategies, load balancing in distributed systems will be more efficient, secure, and scalable.

1. Introduction

Distributed systems, as the core of modern computing architecture, are widely used in fields such as cloud computing, microservices, and data centers. It achieves high availability, scalability, and fault tolerance through multi node collaboration. However, with the expansion of system scale and heterogeneity of resources, how to allocate system load reasonably, prevent resource surplus and performance degradation, has become a key issue in designing distributed systems. Load balancing technology is aimed at improving system performance, accelerating response speed, and processing capabilities through scientific allocation of tasks and resources. In traditional load balancing methods, problems such as uneven load distribution, unpredictable dynamic load changes, and delayed recovery in case of node failures are often encountered, which seriously constrain the overall performance and reliability of distributed systems.

2. The Importance of Load Balancing in Distributed Systems

Load balancing plays an important role in distributed systems, directly affecting the resource utilization, performance, availability, and security of the system. In a distributed system, multiple nodes cooperate with each other to achieve task processing, which requires the system to effectively balance the workload among various nodes. If the load distribution is uneven, it may result in some nodes being overloaded while others are idle, which not only wastes resources but also affects overall performance. Reasonable load balancing can ensure that the resources of each node are fully utilized, thereby improving the overall throughput and response speed of the system.

Load balancing is crucial for improving the high availability and fault tolerance of a system. A major highlight of distributed systems is their excellent high availability, which allows the entire system to operate as usual even if individual nodes fail. Load balancing monitors the status of nodes in real time, automatically reallocating workloads to other normal nodes when nodes fail or encounter problems, ensuring the continuous operation of the system without interruption. In this way, load balancing improves the fault tolerance of distributed systems, enabling the system to maintain stability even in the face of node failures.

Load balancing also supports the scalability of distributed systems. With the continuous growth of business volume, the system often needs to add more new nodes. The load balancing mechanism can dynamically adjust load allocation based on the resource situation of newly added nodes, ensuring that the newly added nodes can efficiently integrate into the existing system and maintain harmonious operation with existing nodes, effectively avoiding system overload or bottlenecks. In addition, load balancing can also play a role in the face of external attacks, such as when encountering DDoS attacks, by dispersing traffic to different nodes to avoid individual nodes bearing too much pressure, thereby ensuring system stability and security.

3. The problems of load balancing in distributed system architecture

3.1. The Complexity of Load Balancing Algorithms

Table 1. The reasons and impacts of the complexity of load balancing algorithms

Problem	Reason	Effect
Resource heterogeneity	Differentiation of node resources (computing power, storage capacity)	It is necessary to comprehensively consider the resource constraints of different nodes, making the algorithm more complex
The system has a large scale	The number of nodes and task volume have increased dramatically, making load scheduling tasks heavy	Traditional load balancing algorithms are unable to efficiently handle large-scale systems and have limited performance
Dynamic load changes	System load changes are influenced by multiple factors and unpredictable	Algorithms require real-time adjustment of load distribution, which increases the difficulty of real-time computation and scheduling
Diverse types of tasks	Different tasks have different computing requirements, priorities, and resource consumption	Multi dimensional scheduling strategies need to be considered, which leads to the complexity of load distribution algorithms
Algorithm calculation cost	Complex algorithms require more computing and storage resources to maintain system efficiency	Increasing system resource consumption reduces the real-time performance and response speed of the system
Real time requirements	The system load fluctuates frequently, requiring quick response and scheduling decisions	Delay increases, making it difficult to handle sudden loads in a timely manner and affecting system stability

With the expansion of distributed systems and the diversification of task types, traditional load balancing algorithms are facing unprecedented challenges. Early load balancing algorithms such as polling and minimum connections relied solely on basic rules to complete load allocation.

As shown in Table 1, with the increase in the number of nodes, heterogeneity of resources, and complexity of tasks, these algorithms gradually show a downward trend in efficiency and accuracy. When dealing with complex systems, algorithms must take into account multidimensional factors, including task priority, node status, resource requirements, etc., which undoubtedly increases the complexity of load balancing algorithm design. In addition, the real-time requirements of load balancing algorithms are higher, especially in scenarios where the load dynamically changes. Traditional algorithms may not be able to respond to load changes in a timely manner, which can affect system performance.

3.2. Dynamic load changes are difficult to predict

In distributed systems, load changes are the result of many factors working together, including the number of user requests, data traffic, network latency, node failures, and so on.

Table 2. Reasons and Effects of Dynamic Load Changes

Problem	Reason	Effect
Sudden traffic	Changes in user behavior, peak activity periods, or increased business demands	Sudden fluctuations in traffic are unforeseeable and may result in some nodes being overloaded or tasks being delayed
Node failure	Node hardware failure, software crash, or network interruption	The load imbalance caused by node failure may not be restored in a timely manner, resulting in performance degradation
Task priority change	The priority changes of different tasks make it difficult for the system scheduling strategy to respond in real-time	High priority tasks may lead to resource competition among low priority tasks, resulting in imbalance
External influences	Load fluctuations caused by factors such as network latency, bandwidth fluctuations, and unavailability of third-party services	The instability of network and service quality increases the difficulty of load prediction
Load trend	System load changes over time, such as seasonal business fluctuations, long-term growth, etc	The changes in long-term load trends affect the accuracy of predictions, and traditional algorithms are difficult to adapt to
Insufficient load forecasting model	The existing prediction models lack adaptability and are unable to handle complex dynamic changes	The model cannot be updated in real-time, resulting in a decrease in the accuracy and timeliness of load forecasting

As shown in Table 2, load changes are usually sudden and highly uncertain. Traditional load balancing algorithms often fail to accurately predict such fluctuations, which may result in the system being unable to make quick adjustments, leading to some nodes being overloaded while others have excess resources. This not only affects the performance of the system, but may also lead to resource waste and a decrease in service quality. More importantly, the complexity of predicting load changes is due to the fact that the load is not only affected by instantaneous factors, but also by the superposition of long-term development trends and seasonal changes, which undoubtedly increases the difficulty of prediction work.

3.3. Malicious attacks leak data privacy

In the context of the widespread application of distributed systems, security issues, especially malicious attacks and data privacy breaches, have become important challenges in system design.

Table 3. Reasons and impacts of malicious attacks and data privacy breaches

Problem	Reason	Effect
Various attack methods	Malicious attack methods continue to evolve, such as DDoS, MITM, SQL injection	Attackers can disrupt load balancing in various ways, leading to system performance degradation or interruption
Trust issues between nodes	The trust chain between distributed system nodes is weak, and malicious nodes may exist	Malicious nodes may tamper with load information or steal sensitive data, disrupting the normal operation of the system
Data transmission is not secure	Data transmission in the network lacks encryption protection, making it easy to be stolen	Sensitive data (user privacy, payment information) may be leaked, causing significant security incidents
Weak security mechanism	Insufficient system security design, lack of defense mechanisms, or failure to promptly patch vulnerabilities	Attackers can easily invade the system, gain unauthorized access, and affect system stability
Load balancing relay problem	During the load balancing process, some information (load status, task allocation) may be leaked	After obtaining this information, attackers can carry out precise attacks, resulting in damage to system functionality
privacy disclosure	Lack of effective privacy protection measures, data storage and processing do not comply with privacy protection regulations	User data leakage may result in legal liability, loss of brand reputation

As shown in Table 3, attackers use various means, such as distributed denial of service attacks (DDoS), man in the middle attacks (MITM), data tampering to interfere with the effective execution of load balancing algorithms and potentially steal critical information within the system. In distributed systems, data transmission and storage are often distributed among multiple nodes, which poses higher requirements for data security maintenance. Once an attacker is able to break through the system's security protection mechanisms, they may gain access to user data, leading to significant impacts such as personal information leakage and service disruptions.

4. Optimization of Load Balancing Technology in Distributed Systems

4.1. Adopting adaptive load balancing algorithm

Adaptive load balancing algorithm is a technology that dynamically adjusts load distribution based on real-time changes system load. Traditional load balancing methods (polling, minimum connections, etc.) often assume that load changes are static or predictable, but load fluctuations in distributed systems are highly uncertain. The adaptive load balancing algorithm can dynamically adjust task allocation strategies through real-time monitoring and prediction, effectively responding to the rapidly changing system load, optimizing resource utilization efficiency, and preventing system overload or ineffective resource consumption.

The core concept of adaptive load balancing is to schedule loads based on the current state of nodes (such as CPU usage, memory usage, bandwidth utilization, etc.) and the characteristics of tasks (such as computational intensity, I/O requirements, etc.). The algorithm collects system status information, analyzes load distribution, and makes intelligent decisions based on it. For example, when the load on a certain node exceeds the threshold, the system will automatically redirect new tasks to nodes with relatively lower loads, or transfer existing tasks to nodes with lighter loads, in order to achieve load balancing of the entire system.

A typical adaptive load balancing formula is as follows:

$$L_i = \frac{T_{total}}{n} \cdot f(node\_status_i, task\_type_i) \quad (1)$$

Among them, L_i represents the load of the i th node, T_{total} represents the total task load, and $f(\cdot)$ is an adaptive function that dynamically adjusts load allocation considering the status of nodes and task types.

The significant advantage of adaptive load balancing algorithm lies in its ability to perform real-time optimization based on the current operating status of the system, overcoming the limitations of traditional static load allocation algorithms. However, the main challenge it faces is how to cope with real-time challenges, especially when the system scale is large, ensuring the efficiency and accuracy of algorithm operation is particularly crucial. In addition, the computational cost of algorithms is also an important issue that cannot be ignored. In the complex environment of distributed system structure, maintaining the response time and stability of the system becomes the core of optimization work.

4.2. Utilizing machine learning and deep learning models

In the field of load balancing, the integration of machine learning and deep learning has brought more advanced intelligent solutions for dynamic system adjustment. By utilizing historical load data and real-time monitoring data, machine learning models can effectively predict changes in system load and intelligently optimize task allocation based on this. Compared with traditional methods, machine learning models have higher adaptability and accuracy, and are able to flexibly respond to sudden changes in load and complex system dynamics.

Common machine learning methods, including support vector machines (SVM), decision trees, and linear regression, can construct predictive models through in-depth analysis of historical data (such as past load distribution, task types, system states, etc.), providing decision-making references for future load allocation. For example, using regression analysis, machine learning can predict the future load of a node and make scheduling decisions in advance. Assuming that the load characteristic of each task is x, and the output of the machine learning model is the predicted load $\widehat{L_i}$:

$$\widehat{L_i} = f(x_i) \tag{2}$$

Among them, x_i contains historical data, node information, etc. of the task. $f(\cdot)$ is a machine learning model that outputs predicted values for future loads.

The use of Long Short Term Memory (LSTM) networks, a deep learning technique, can effectively process time series data and is suitable for capturing long-term dependencies of load fluctuations. By training on historical load data, the LSTM network can accurately predict future load fluctuations and optimize task allocation strategies accordingly. The formula for LSTM network is as follows:

$$\widehat{L_{t+1}} = LSTM(L_{t-k}, L_{t-k+1}, \dots, L_t) \tag{3}$$

Among them, L_t is time series data, k is historical time window, and $LSTM$ can predict future load by learning the pattern of load changes. Through machine learning and deep learning, the system can accurately improve the accuracy of load prediction and autonomously optimize load allocation schemes in changing environments, thereby achieving efficient resource allocation and comprehensive system optimization.

4.3. Implementing secure load balancing for blockchain

In distributed systems, load balancing technology is used to ensure that workloads are evenly distributed among multiple nodes, prevent individual nodes from being overloaded, and thus improve the overall operational efficiency of the system. However, as the scale of the system continues to expand, traditional load balancing techniques face significant challenges in terms of security, such as malicious attacks, data breaches, and task tampering. Blockchain technology, with its decentralization, data immutability, and transparency, provides a new approach to solving the security challenges of distributed systems. The decentralized nature of blockchain makes the load balancing process more transparent and avoids the risk of single point of failure. In the load balancing scheme using blockchain, the status information of each node (such as current load, processing capacity, resource usage, etc.) and the historical records of load allocation will be written into the blockchain ledger. In this way, the task allocation of each node can be checked and verified in real time, ensuring the fairness and tamper resistance of load distribution.

The load distribution of tasks can be automatically executed through smart contracts. Smart contracts are pre written and deployed programs on the blockchain, which automatically execute task scheduling once preset conditions are triggered. For example, once the burden on a node exceeds the threshold, the smart contract will actively assign subsequent tasks to nodes with lighter burdens to achieve reasonable resource allocation. Assuming there are n nodes in the system, each with a processing capacity of P_i, a load status of L_i, and a total task load of T_{total}, task allocation can be represented by the following formula:

$$L_i = smart\_contract(P_i, node\_status_i) \tag{4}$$

Among them, $smart\_contract(\cdot)$ represents smart contracts, which automatically adjust task allocation based on the processing capacity and load status of nodes. By leveraging the decentralized nature of blockchain technology and smart contract mechanisms, the fairness of task allocation can be ensured, the risk of malicious behavior and data tampering can be avoided, and the scalability and fault tolerance of the system can be enhanced.

5. Conclusion

In distributed system architecture, implementing load balancing technology is a key step in ensuring efficient and stable system operation. With the continuous development of technology, traditional load balancing methods face multiple challenges such as dynamic load changes, computational complexity, and security. To address these issues, advanced technologies such as adaptive algorithms, machine learning, deep learning, and blockchain can significantly enhance the intelligence and security of load balancing. The introduction of blockchain is particularly significant, as its decentralized recording method and the application of smart contracts greatly enhance the transparency, fairness, and security of load distribution. By continuously optimizing the load balancing strategy, distributed systems can not only adapt more effectively to load fluctuations, but also enhance their fault tolerance and attack resistance.

Reference

[1] Patel H, Rajput D S, Stan O P, et al. A New Fuzzy Adaptive Algorithm to Classify Imbalanced Data. Computers, Materials, and Continues (in English), 2022(1):73-89.

[2] Fateh R, Darif A, Safi S. An Extended Version of the Proportional Adaptive Algorithm Based on Kernel Methods for Channel Identification with Binary Measurements. Journal of Telecommunications and Information Technology, 2022(3):47-58.

[3] Yu T, Wang T, Chen H, et al. An Integrated Load Balancing Scheme Based on SDN ASR Algorithm. China Communications: English version, 2023, 20(3):9.

[4] Rajeh W. Hadoop Distributed File System Security Challenges and Examination of Unauthorized Access Issue. Journal of Information Security, 2022, 13(2):23-42.

[5] Hongvanthong S, Chunlin L. A novel four-tier software-defined network architecture for scalable secure routing and load balancing. International journal of communication systems, 2022(1):35.

[6] Su H, Luo W, Mehdad Y, et al. Llm-friendly knowledge representation for customer support[C]//Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025: 496-504.

[7] Lu, C. (2025). Application of Multi-Source Remote Sensing Data and Lidar Data Fusion Technology in Agricultural Monitoring. Journal of Computer, Signal, and System Research, 2(7), 1-6.

[8] Ye, J. (2025). Optimization of Neural Motor Control Model Based on EMG Signals. International Journal of Engineering Advances, 2(4), 1-8.

[9] Liu, Y. (2025). Use SQL and Python to Advance the Effect Analysis of Financial Data Automation. Financial Economics Insights, 2(1), 110-117.

[10] Sun, Q. (2025). Research on Cross-language Intelligent Interaction Integrating NLP and Generative Models. Engineering Advances, 5(4).

[11] K. Zhang, "Optimization and Performance Analysis of Personalized Sequence Recommendation Algorithm Based on Knowledge Graph and Long Short Term Memory Network, " 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.

[12] J. Huang, "Research on Multi-Model Fusion Machine Learning Demand Intelligent Forecasting System in Cloud Computing Environment, " 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7.

[13] F. Liu, "Transformer XL Long Range Dependency Modeling and Dynamic Growth Prediction Algorithm for E-Commerce User Behavior Sequence," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-6.

[14] Zou, Y. (2025). Design and Implementation of a Cloud Computing Security Assessment Model Based on Hierarchical Analysis and Fuzzy Comprehensive Evaluation. arXiv preprint arXiv:2511. 05049.

[15] Zhu, P. (2025). The Role and Mechanism of Deep Statistical Machine Learning In Biological Target Screening and Immune Microenvironment Regulation of Asthma. arXiv preprint arXiv:2511. 05904.

[16] Ding, J. (2025). Intelligent Sensor and System Integration Optimization of Auto Drive System. International Journal of Engineering Advances, 2(3), 124-130.

[17] Mingjie Chen. (2025). Exploration of the Application of the LINDDUN Model in Privacy Protection for Electric Vehicle Users. Engineering Advances, 5(4), 160-165.

[18] Liu, X. (2025). Research on Real-Time User Feedback Acceleration Mechanism Based on Genai Chatbot. International Journal of Engineering Advances, 2(3), 109-116.

[19] Zhang, M. (2025). Research on Collaborative Development Mode of C# and Python in Medical Device Software Development. Journal of Computer, Signal, and System Research, 2(7), 25-32.

[20] Wang, Y. (2025). Intervention Research and Optimization Strategies for Neuromuscular Function Degeneration in the Context of Aging. Journal of Computer, Signal, and System Research, 2(7), 14-24.