Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33733

Overview of Cloud Side Collaborative Computing Architecture and Algorithm

Zhenxuan Zhang

Langfang Branch of Hebei University of Technology, Langfang, China
Lucas.Zhang.1@uni.massey.ac.nz

Abstract. In the face of massive data processing requirements in the era of Internet of Things (IoT) and big data, traditional centralized cloud computing has obvious limitations in terms of latency, bandwidth, and reliability. Cloud-edge-end collaborative computing is based on the terminal-edge-cloud three-level architecture, which effectively reduces latency and power consumption and improves system stability through hierarchical task processing and computing power collaboration. This paper focuses on the cloud-edge-end architecture, elaborates on the roles of the terminal layer, edge layer and cloud layer, shows the three layers of closely coordinated and indispensable framework system, compares the differences between the cloud-edge-end framework and the traditional framework, and derives the importance of clock synchronization technology in the framework with the application of distributed computing in the framework. This paper expounds the distributed core technology of fault tolerance mechanism, and finally summarizes the challenges and future research trends of cloud-edge-end architecture in various fields.

Keywords: Cloud side architecture, Distributed architecture, Clock synchronization, Fault tolerance mechanism

1. Introduction

With the rapid development of high-tech technologies such as the Internet, 5G communication, and AI, the number of IoT terminal devices and data volume has increased exponentially. Secondly, the privacy of users is becoming more and more important, the increase in the number of terminals has generated a large amount of data, which often contains a large amount of personal information, however, in the process of uploading these data from the terminal layer to the cloud server, the user's privacy will have a huge risk of leakage, in addition, the data uploaded to the edge layer and the cloud may be at risk of being forged, but it is difficult for the edge layer and the cloud layer to identify the false data in it, which may eventually lead to inaccurate control and decision-making. Causing serious economic losses [1], and finally, data centers and their computing services, as one of the main forces of global energy consumption today, if the information is simply concentrated in the cloud server for centralized computing, the computing resources of the terminal equipment will be wasted, which will not be conducive to energy conservation and reduce the power consumption of the server [2].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU33733

In order to solve the above problems, a cloud-edge-end collaborative computing architecture has been proposed, which realizes the unified deployment and complementary advantages of cloud computing, edge computing and terminal devices, significantly reduces system latency and device power consumption, and alleviates the pressure of network bandwidth. Enhanced security in terminal information. This paper starts from the basic architecture of the cloud-edge-end and compares it with the traditional centralized architecture, highlighting the many advantages of the cloud-edge-end collaborative computing architecture, and then expounds the core algorithm distributed algorithm of the cloud-edge-edge architecture, focusing on the advantages and applications of each technology in the clock synchronization technology and fault-tolerant mechanism. And look forward to future research directions.

2. Basic layered architecture

2.1. Cloud edge and architecture

The cloud-edge-end collaborative computing architecture is a computing model with the three-level layering of the terminal layer, edge layer, and cloud layer as the core architecture The computing system is formed through key links such as network resource interconnection, task division, and unified coordination and scheduling. Among them, the number of terminal layer devices is huge and diverse, this layer often collects massive data, generating huge computing tasks, these tasks are often real-time tasks, need to complete the corresponding within strict time limits, and the terminal itself has a certain computing power, but most of its performance cannot meet the demand, so the terminal equipment often forwards the resource requirements to the server; The edge layer is deployed in large numbers at the edge of the network, with a certain computing power, which makes up for the lack of terminal computing power, and can respond to terminal requests in a timely manner, greatly reducing the latency of end users. The cloud layer adopts centralized deployment and is composed of large-scale high-performance server clusters, which makes the cloud have extremely powerful computing power and huge storage space, but the deployment location of the cloud is often far away from the terminal equipment, so it has long latency and large bandwidth pressure when network transmission is carried out [3]. Through the design of nearby computing and global collaboration, computing power and data can be allocated on demand, so that the low latency advantages of the edge layer can be exerted, and the ultra-large-scale computing power of the cloud layer can be utilized. Overall, relying on this cloud-edge-end three-layer architecture, it effectively avoids the limitations of traditional computing architectures and allows the Internet of Things, intelligent transportation, and other fields to be fully developed [2].

2.2. Comparison between cloud side architecture and traditional centralized cloud computing architecture

The cloud-edge-end collaborative architecture realizes the complementary advantages of computing power and functions, and solves the pain points of traditional computing models. As shown in Table 1, traditional centralized cloud computing adopts a terminal-central cloud architecture to converge all computing tasks in the cloud for processing, although it can rely on massive computing power to complete complex calculations, but there are problems such as high latency and large bandwidth consumption, which are difficult to apply to low-latency scenarios such as industrial control and autonomous driving. The cloud-edge-end collaboration architecture adopts a three-layer hierarchical processing framework of terminal-edge-cloud, through the division of tasks, reasonably and

DOI: 10.54254/2755-2721/2026.GU33733

efficiently assigns tasks to each level, the collection and preprocessing of information at the terminal, the nearby real-time processing of the edge, and the large-scale parallel computing of the cloud, so that it has the characteristics of low latency, significantly reduced bandwidth pressure compared with traditional computing frameworks, and high reliability, and is now widely used in high real-time performance such as autonomous driving and the Internet of Things , fields with high reliability requirements.

Table 1. Comparison of centralized computing architecture and cloud-edge edge architecture

	Traditional centralized computing architecture	Cloud-edge-edge collaboration architecture
Architecture pattern	Terminal-Center Cloud	Terminal layer - edge layer - cloud layer
Data processing	All data depends Central cloud processing	Terminal: Pretreatment Edge: Real-time processing Cloud: global analysis and computing
Latency and real-time	High latency and does not support real-time data processing	Low latency and real-time processing at the edge
Network dependency	Dependent on the network, the network is disconnected and unavailable	Low dependency, can still operate when the network is disconnected
Network broadband pressure	The original data is fully uploaded, and the broadband pressure is high	Edge data preprocessing significantly reduces broadband pressure
System reliability	Poor reliability	High reliability
Adapt to the scenario	Not real-time, centralized	Low-latency scenarios such as IoT and industrial control

3. Distributed algorithm

The cloud-edge-end collaborative framework is essentially the application of a distributed framework, which disperses information collection, storage, task processing and other capabilities to terminals, edges and clouds, from the consistency of time benchmarks between nodes, to the stability of the system when faults occur, and then to the reasonable allocation and offloading of computing tasks. The distributed computing system is composed of a large number of heterogeneous nodes, so in order to prevent the system crash error caused by the error of the execution order or the deadline of each module, the clock synchronization technology plays an indispensable role, among which the Lamport clock algorithm is the cornerstone of the clock synchronization technology, which is of milestone significance for the future development of this technology. On this basis, the three clock alignment schemes of hardware-triggered synchronization, software timestamp matching, and message filter-based synchronization further expand the accuracy and scenario adaptability of time synchronization, and at the same time, the fault-tolerant mechanism and consensus algorithm are supported, so that the distributed system has both efficiency and reliability, so that it can be applied in various complex fields.

3.1. Clock synchronization

If people want the distributed system to function properly, people need to keep the clock consistent among the nodes in the system, so it is used to ensure the reliability of the consistent time and event sequence between distributed nodes. According to the real-time requirements, NTP (millisecond precision), PTP (microsecond precision), satellite clock, or distributed synchronization algorithms

DOI: 10.54254/2755-2721/2026.GU33733

can be used to achieve synchronization with different precisions [4]. Therefore, the accuracy of clock synchronization directly determines the reliability of distributed collaboration, and the accuracy of time synchronization is affected by a variety of factors, mainly affected by three aspects: protocol stack level delay, network environment disturbance, and hardware and environmental noise [5], Ref [4]. The proposed PTP clock synchronization parameter tracking method based on alternating minimization Kalman filter effectively slows down the influence of the above factors on clock accuracy.

3.1.1. Lamport clock

In some distributed systems with low accuracy requirements, the Lamport clock correction algorithm can be used for synchronization processing, and when using the Lamport logic clock, it is important to adhere to the important premise that two events cannot occur exactly at the same time. The Lamport clock establishes the causal sequence of distributed events "happens before", which is the basis for understanding the logical clock, which does not require additional hardware support and relies entirely on software logic, which can ensure that the causal dependent events have strictly incremental logical timestamps.

The Lamport clock can achieve causal sequencing of events through three simple formulas, one of which is $C_i = C_i + 1$,That is, when a local event occurs, the local clock value is added by 1;The second is $C_i = C_i + 1$, and then send, that is, when a message sending event occurs, add the local clock value by 1, and then send the updated clock value with the message; the third is $C_i C_j = \max(C_i, C_j) + 1$,That is, when a message receiving event occurs, the greater of the sender's clock value and the local clock value is taken first, and then the result is added by 1 as the new local clock value [6].

3.1.2. Three clock alignment schemes

The hardware trigger synchronization scheme uses a unified hardware signal (such as PPS) to directly trigger all sensors to collect data at the same time, realizing the matching from the physical level, and the accuracy of this scheme is very high, and the error can be less than 1μs. In addition, this solution is highly dependent on hardware and requires sensors to have trigger pins. As a result, costs are unavoidably higher than other alignment methods.

In this scheme, each frame of data is entered into a high-precision PTP hardware timestamp, and then matched to a unified time grid through an interpolation algorithm on the CPU side. Its advantage is high flexibility and does not rely on special hardware; However, the accuracy of this scheme is limited by timestamp resolution and CPU jitter, usually in the millisecond range.

Synchronization based on message filters uses ROS 2's "message_filters" tool to subscribe to multiple topics at the software layer and trigger a fusion callback only when the timestamps are close enough. Therefore, this method does not need to modify the hardware device, only needs to set a reasonable tolerance, which is a powerful tool for quickly verifying the fusion algorithm, but it needs to pay attention to the jitter of the CPU under high load.

3.2. Fault tolerance mechanism

Fault tolerance refers to the ability of the system to maintain the normal operation of core services when some components fail, software process crashes, data abnormalities and other failures, and its core is embodied in the immune characteristics to resist the spread of local faults to the whole

DOI: 10.54254/2755-2721/2026.GU33733

system, as well as the self-healing characteristics that can complete fault identification and service recovery without manual intervention. Its core connotation is not to pursue absolute zero failure of the system, but to ensure that the system can continue to provide core services when a fault occurs, minimizing service interruption, data loss, and user experience impact caused by failures [7].

The core value of the fault tolerance mechanism is reflected in two dimensions: ensuring service continuity and reducing the operation and maintenance costs of the entire life cycle of the system. Financial transactions, medical and other scenarios need to ensure uninterrupted service within 24 hours, if the operation of the server is interrupted, it will cause huge economic losses or security accidents; The automatic fault detection and recovery mechanism can greatly reduce the investment of human resources, and greatly reduce the maintenance cost of long-term server operation compared to systems designed without fault-tolerance [7].

Redundant backup technology is mainly divided into two types: data copy redundancy and service redundancy.

Data replica redundancy is designed to ensure that data is not lost through multi-replica storage [8]. Taking MySQL master-slave replication as an example, the core of this technology is that the master server records all data changes to the binary log, and then reads the log from the server through the I/O thread and writes the relay log, and then the SQL thread replays the log events to achieve data synchronization, and finally realizes data synchronization with the master database.

MySQL master-slave replication provides three modes: asynchronous replication, semi-synchronous replication, and synchronous replication. Asynchronous replication is the one in which the main library writes to the binlog and returns to the client immediately without waiting for confirmation from the database. Semi-synchronous replication is a case where the primary database writes to the binlog and waits for at least one slave to write the log to the relay log and return it for confirmation before returning to the client, striking a balance between performance and data consistency, effectively reducing the risk of data loss. Synchronous replication requires the main database to wait for all slaves to complete data replay and confirm before returning to the client, which can ensure strong consistency but has low performance and is only suitable for scenarios with extremely high data consistency requirements. These three master-slave replication technologies based on binary logs often have latency problems, single point of failure problems, and data consistency problems, which also make these modes ineffective in today's places with high requirements for real-time and flexibility. The node scheduling mechanism and its enhanced fault resilience are introduced, which significantly improves the performance and reliability of MySQL master-slave replication [9].

When a node fails, goes down or is overloaded, the system can automatically switch the request to a normal node to ensure that the service is not interrupted and the calculation does not stop, and avoids the overall system paralysis caused by a single point of failure [10].

4. Conclusion

The cloud-edge-end three-level collaboration architecture fundamentally solves the problems of high latency, high bandwidth occupancy, and low reliability of traditional centralized computing, and allocates computing power with nearby computing and global scheduling. Clock synchronization ensures the consistent timing of distributed nodes, and the fault tolerance mechanism ensures that the system can operate stably in the event of partial failures, which are the key technical support for cloud-edge-end collaboration. In addition, distributed computing has entered a period of stable development, and the hot spots are concentrated in big data, cloud platforms, and AI integration; In the future, it is necessary to strengthen cooperation, deepen interdisciplinary research, and improve

DOI: 10.54254/2755-2721/2026.GU33733

security, privacy and resource scheduling capabilities. With a large number of terminals and edge nodes, and the pressure on data security and privacy protection is prominent, how to reduce the risk of user privacy leakage is also an important challenge.

References

[1] Wang, Y.J., Wang, Z.Y. (2025) Analysis of the Domestic Development Trend of Distributed Computing. Modern Information Science and Technology, 9(13), 170-174.
[2] Fang, W.D., Zhu, C.S., Zhang, W.X. (2024) Toward Secure and Lightweight Data Transmission for Cloud–Edge–Terminal Collaboration in Artificial Intelligence of Things. Unknown Journal, 11, 105-113.
[3] Dong, Y.M., Zhang, J., Xie, C.Z., et al. (2024) A Review of Key Issues of Edge Intelligent Computing under Cloud-Edge-End Architecture: Computing Optimization and Computing Offload. Journal of Electronics and Information Technology, 46(3), 765-776.
[4] Yu, W., Liang, F., He, X.F., Hatcher, W.G., Lu, C., Lin, J., Yang, X.Y. (2018) A Survey on the Edge Computing for the Internet of Things. Unknown Journal, 6, 6900-6919.
[5] Wang, H., Wang, H., Liu, X.J. (2026) PTP Clock Synchronization Parameter Tracking Method Based on Alternating Minimization Kalman Filter. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 1-10. (Online first)
[6] Xing, X.R. (2025) Research on Wireless Time Synchronization Technology in Heterogeneous Networks. Beijing University of Posts and Telecommunications. (Doctoral dissertation)
[7] Wei, J.J. (2004) Title unknown. Aviation Computing Technology, (3), 113-115. (Note: original record missing article title)
[8] Xu, M. (2012) Research and Implementation of Fault Tolerance Mechanism of Distributed File System. University of Electronic Science and Technology of China. (Master's thesis)
[9] Huang, S.J. (2025) Collaborative Optimization of Data Redundancy and Fault Tolerance Mechanism in Network Big Data Distributed Storage System. In Proceedings of the 10th Academic Exchange Conference on Engineering Technology Management and Digital Transformation, 391-393.
[10] He, Q.Y., Chen, J., Fang, G.C. (2025) Research on MySQL Database Master-Slave Replication Reliability Technology. Mechanical and Electrical Engineering Technology, 54(12), 123-128.