A Review of the Evolution and Challenges of the Collaborative Architecture of Edge Computing and Cloud Computing

Yuxuan Dong

Xi'an Technological University, Xi’an, China

2174207343@qq.com

Abstract. With the deep integration and large-scale application of Internet of Things (iot), 5G communication and artificial intelligence technologies, global data is experiencing explosive growth. The limitations of traditional centralized cloud computing architectures in terms of ultra-low latency service demands, massive bandwidth consumption and data privacy protection are becoming increasingly prominent. Edge computing, as an important extension of distributed systems, effectively compensates for the shortcomings of cloud computing in real-time response and bandwidth optimization by sinking computing, storage and application capabilities to the network edge. However, the inherent characteristics of edge nodes, such as limited computing power and heterogeneous resources, make it difficult for them to independently undertake large-scale global optimization tasks. Therefore, "cloud-edge collaboration" has become the core paradigm to solve the contradiction of computing services in the Internet of Things era. From the perspective of distributed system theory, this paper systematically sorts out the evolution context of the cloud-edge collaborative architecture, and deeply analyzes the three-layer logical model of "endpoint device layer - relay edge layer - core cloud computing layer" and its internal collaborative mechanism. Focus on researching key technologies such as computing offloading decision optimization, data consistency maintenance, cloud-edge resource orchestration and communication, and construct multi-objective optimization models and efficient solutions; Conduct empirical analysis based on typical scenarios such as smart cities, Industry 4.0, and intelligent connected vehicles to verify the technical superiority of the cloud-edge collaborative architecture. Finally, the future research directions such as green edge computing, endogenous security mechanisms, and AI adaptive collaboration are prospected. The research results can provide references for the theoretical innovation and engineering implementation of cloud-edge collaborative systems.

Keywords: Cloud-edge collaboration; distributed system; Calculate the offload; Data consistency

1. Introduction

Object detection, as one of the core tasks in computer vision, aims to simultaneously locate objects within an image and identify their categories. Unlike image classification or object recognition, object detection not only determines “what it is” but also answers “where it is.” Consequently, object detection holds critical application value in fields such as video surveillance, autonomous driving, smart healthcare, facial recognition, and industrial quality inspection, serving as a vital foundation for intelligent vision systems.

In the context of the rapid development of the digital economy, cloud computing has become the mainstream architecture paradigm of distributed systems in the past decade with its core advantages such as resource pooling, on-demand allocation, elastic expansion, and low-cost maintenance. The widespread application of cloud service platforms such as Amazon AWS, Microsoft Azure, and Alibaba Cloud has promoted the large-scale development of the Internet, big data and artificial intelligence industries, and provided core support for the digital transformation of enterprises. However, with the commercial deployment of 5G networks and the explosive growth of IoT devices, the data production model has undergone fundamental changes, and the traditional "end-cloud" centralized architecture is facing unprecedented challenges.

In recent years, cloud-edge collaboration technology has become a research hotspot. Scholars and enterprises at home and abroad have conducted extensive research in aspects such as architecture

https://doi.org/10.58244/joerai.263732

design, key technologies, and application scenarios. In terms of architectural design, the industrial sector has proposed solutions such as Huawei's "cloud-edge-device" three-layer architecture, Alibaba Cloud's LinkEdge platform, and Microsoft's AzureIoTEdge. Meanwhile, the academic community has focused on issues like heterogeneous node collaboration and dynamic topology adaptation, and has put forward various architectural models including hierarchical and distributed models. In terms of computing offloading technology, existing research mainly adopts methods such as optimization theory, machine learning, game theory and heuristic algorithms to achieve the optimal allocation of tasks between the cloud and the edge. However, most of them lack the adaptive ability to dynamic network environments and heterogeneous resources. In terms of maintaining data consistency, scholars have made improvements based on traditional protocols such as Paxos and Raft, or adopted eventual consistency protocols, but it is difficult to balance data accuracy and system performance. In terms of resource orchestration and communication, open-source projects such as KubeEdge and EdgeXFoundry have achieved unified management of cloud-edge resources. Lightweight protocols like MQTT and CoAP are widely used in weak network communication, but the efficiency and flexibility of resource scheduling still need to be improved.

Although certain progress has been made in existing research, there are still many deficiencies: the lack of a unified standard for architecture design and the difficulty in interconnection and interoperability among different platforms; The dynamic adaptability of the computing offloading algorithm is insufficient, making it difficult to cope with complex and changeable application scenarios. The balance mechanism between data consistency and system performance is not yet perfect. Resource orchestration does not fully consider the heterogeneity and dynamics of edge nodes. The security protection mechanism is difficult to adapt to edge resource-constrained scenarios. Therefore, it is urgently necessary to conduct more in-depth research and build an efficient, reliable and secure cloud-edge collaboration system.

2 In-depth Analysis of Cloud-Edge Architecture

2.1 Evolution of Distributed Systems

The evolution of distributed systems has experienced four key stages: the early master-slave architecture has a simple structure but suffers from server bottlenecks and single point of failure problems; the peer-to-peer network architecture features a high degree of decentralization and strong scalability, but low resource scheduling efficiency and poor security; the cloud computing architecture realizes elastic expansion through resource pooling and reduces operation and maintenance costs, but faces bandwidth and real-time bottlenecks; the cloud-edge collaborative architecture integrates the advantages of cloud computing and edge computing, solving the above contradictions through hierarchical collaboration and becoming the mainstream development direction of distributed systems.

The core characteristics of the cloud-edge collaborative architecture are hierarchical computing, heterogeneous collaboration and dynamic scheduling: the edge layer focuses on real-time task processing, the cloud layer on global task processing, and the terminal is responsible for data collection and simple execution. The three realize optimal resource allocation and collaborative task execution through efficient communication, forming a closed-loop mode of "edge perception and response, cloud decision-making and optimization".

2.2 Three-Layer Logical Architecture Model of Cloud-Edge Collaboration

Based on distributed system theory, this paper proposes a three-layer logical architecture model of "end device layer - relay edge layer - core cloud computing layer". The functional positioning and characteristics of each layer are as follows:

1) End Device Layer. As the source of data generation, this layer consists of various IoT terminals distributed in geographic space, including sensors, cameras, smart wearable devices, industrial controllers and on-board terminals. From a distributed perspective, the nodes in this layer are data "producers", and their core functions include: raw data collection, acquiring physical world signals through built-in sensors or external interfaces, and conducting preliminary processing such as analog-to-digital conversion and data filtering; local simple computing, limited by the computing power and

https://doi.org/10.58244/joerai.263732

power consumption of embedded microcontrollers, only capable of executing lightweight tasks such as data compression and simple feature extraction, and some terminals can realize simple reasoning through edge-side AI technology; data transmission, uploading data to the edge layer or cloud layer through wireless or wired networks; instruction execution, receiving control instructions from upper-layer nodes and completing corresponding operations.

The main characteristics of this layer are: a huge number and wide distribution; extremely limited computing power and sensitivity; strong hardware and software heterogeneity, diverse device types and different communication protocols.

2) Relay Edge Layer. As the core innovative layer of the architecture, edge nodes are deployed close to the access network side (such as beside base stations, workshop sites and roads), between terminals and the cloud, acting as the role of "regional coordinator". Edge layer nodes mainly include edge gateways, edge servers, road side units and industrial edge controllers, with core functions including: real-time task processing, using geographic proximity to process delay-sensitive tasks such as real-time preprocessing of video streams, real-time control of industrial equipment and traffic early warning, reducing the end-to-end delay to the millisecond level; data aggregation and preprocessing, receiving data from multiple terminals, eliminating redundant information, extracting key features, compressing and then uploading to the cloud, greatly reducing the transmission pressure of the backbone network; local decision-making and control, issuing control instructions based on local data and preset models to realize localized closed-loop control, improving the system response speed and reliability; cache service, caching frequently accessed data, model parameters and applications issued by the cloud, reducing terminal access delay and bandwidth consumption; offline autonomy, capable of independently completing local task processing in the case of network disconnection, and synchronizing data with the cloud after network recovery.

The main characteristics of this layer are: computing power and storage capacity between terminals and the cloud, which can be flexibly configured; strong geographic proximity and low transmission delay; prominent hardware heterogeneity and topology dynamics, and some nodes have mobility.

3) Core Cloud Computing Layer. Composed of large-scale cloud computing centers, it is the "global brain" of the system, with nearly unlimited elastic computing power and massive storage capacity. Its core functions include: large-scale computing task processing, such as deep learning model training, global big data analysis and cross-regional resource optimization scheduling, realizing global model optimization and local deployment through the mode of "cloud training and edge reasoning"; massive data storage and management, storing key data uploaded by the edge layer, raw terminal data and model parameters, and realizing efficient management and query through distributed storage and database technologies; global decision-making and optimization, analyzing global data and formulating optimization strategies such as urban traffic scheduling plans and industrial production process optimization schemes, and issuing them to the edge layer and terminals; unified resource orchestration, monitoring and scheduling cloud-edge-end full-link resources to realize load balancing and efficient resource utilization; system operation and maintenance and security management, responsible for equipment monitoring, fault diagnosis, software upgrade and security protection.

The main characteristics of this layer are: powerful computing power and elastic expansion; huge storage capacity and high network bandwidth; adoption of a centralized management mode with high management efficiency.

4) Collaborative Mechanism of the Three-Layer Architecture. The three-layer architecture achieves efficient operation through task division, data flow, resource scheduling, model collaboration and security collaboration: in terms of task division, real-time and lightweight tasks are processed by terminals or the edge layer, and large-scale and global tasks by the cloud layer; data flow follows the path of "terminal collection - edge preprocessing - cloud storage and analysis" to achieve bandwidth optimization and processing efficiency improvement; resource scheduling is centrally managed by the cloud, dynamically allocating resources according to node load and task requirements; model collaboration realizes the balance between global intelligence and local adaptation through "cloud training - edge fine-tuning - terminal reasoning"; security collaboration builds a multi-level protection

https://doi.org/10.58244/joerai.263732

system, with terminals ensuring local data security, the edge layer filtering malicious data, and the cloud deploying global security protection.

3 Analysis of Key Technologies in Cloud-Edge Collaboration

3.1 Data Consistency Maintenance Technology

Although the cache mechanism of edge nodes can alleviate the bandwidth pressure, it causes the consistency problem of multi-copy data. Traditional distributed consistency protocols have problems such as high synchronization delay and low network partition tolerance in the weak network environment of cloud-edge collaboration, so it is necessary to design a consistency scheme adapted to the cloud-edge scenario.

1) Hierarchical Lease Consistency Mechanism. This paper proposes a hierarchical consistency scheme based on the lease mechanism: the cloud acts as the manager of the main data copy and grants edge nodes a data modification lease for a specific period. Edge nodes can independently modify local cache copies within the lease validity period and record modification operations; before the lease expires, edge nodes need to apply to the cloud for lease renewal. If the renewal is successful, the modification permission is retained; otherwise, the local modifications need to be synchronized to the cloud and the permission is released. By maintaining the lease state, the cloud ensures that only authorized edge nodes can modify data in the same time period, reducing the probability of conflicts. The core advantages of this mechanism are: moderate synchronization delay, only requiring lease verification and periodic synchronization without cross-network consensus; good network partition tolerance, edge nodes can operate offline during the lease period and synchronize data after network recovery.

2) Conflict Detection and Fusion Based on Vector Clock. When a network partition occurs, multiple edge nodes may modify the same data copy during the lease period, leading to data conflicts. A vector clock is used to record the partial order relationship of data operations. Each edge node maintains a vector $VC=(v_1, v_2, ..., v_n)$, where v_i represents the number of modifications to the data by node i. After the network recovery, the data version relationship is judged by comparing vector clocks: if $VCA \leqslant VCB$, then A is an old version of B, and A is directly overwritten by B; if VCA and VCB are incomparable, there is a concurrent conflict, which needs to be resolved according to preset rules.

Table 1 Performance Comparison of Consistency Models in Cloud-Edge Collaboration Systems

Evaluation Dimension	Strong Consistency (Paxos/Raft)	Eventual Consistency (CRDTs)	Hierarchical Lease Mechanism (Recommended in this paper)
Synchronization Delay	Extremely high (requiring cross-network consensus)	Extremely low (local update)	Moderate (only lease verification required)
Network Partition Tolerance	Poor (prone to service outage)	Excellent (supporting offline operation)	Good (operable during the lease period)
Data Accuracy	Absolutely synchronous	Temporary conflicts exist	Logically ordered

3) Performance Comparison of Consistency Models.

Table 1 compares the performance differences among strong consistency, eventual consistency (CRDTs) and the hierarchical lease mechanism. Strong consistency can ensure absolute data synchronization, but with extremely high synchronization delay and poor network partition tolerance; eventual consistency has low synchronization delay but temporary data conflicts; the hierarchical lease mechanism achieves a good balance among synchronization delay, network partition tolerance and data accuracy, which is more suitable for cloud-edge collaboration scenarios.

3.2 Game Theory and Heuristic Strategies for Computing Offloading

In a distributed cloud-edge system, computing offloading is not only a decision of a single node, but also involves the resource game of multiple users competing for the computing power resources of edge servers.

1) Task Modeling and Multi-Objective Optimization. The attributes of task i are defined as $Task_i=\{D_i, C_i, T_{max}, E_{max}\}$, where E_{max} represents the maximum tolerable energy consumption. In a

distributed environment, the offloading decision $a_i \in \{0,1,2\}$ (representing local, edge and cloud respectively) needs to solve a multi-objective optimization function:
min where α and β are weight factors, representing the application's preference for "real-time performance" or "power saving".
2) Offloading Decision Based on Game Theory and Heuristic Algorithms. When multiple terminals request offloading from edge nodes at the same time, resource competition and channel interference will be caused, and resource allocation needs to be optimized through game theory models. Each terminal is regarded as a participant in a non-cooperative game, and its profit function is defined as task completion efficiency. By solving the Nash equilibrium, the transmission power and computing frequency of each terminal are dynamically adjusted to avoid "request explosion" on the edge side.
Aiming at the high complexity of game theory algorithms in large-scale systems, heuristic algorithms are combined to improve decision-making efficiency. The algorithm steps are as follows: Initialize the population: randomly generate a set of offloading decision schemes (chromosomes), where each chromosome corresponds to the offloading choice of each task. Fitness evaluation: calculate the fitness value of each chromosome based on the multi-objective optimization function. Selection, crossover and mutation: generate a new generation of population through roulette selection, single-point crossover and random mutation. Iteration termination: output the optimal offloading decision when the number of iterations reaches the threshold or the fitness value converges.
3) Logical Flow of Offloading Decision. As shown in Figure 3-1, after a task arrives at the edge gateway, its real-time requirement is first detected: if the delay requirement is extremely high, it is executed directly on the local terminal; if the real-time requirement is medium or high, it is judged whether the edge resources are sufficient. If sufficient, the task is offloaded to the edge for execution; otherwise, the network bandwidth status is detected. If the bandwidth is good, the task is offloaded to the cloud for processing; if the bandwidth is congested, it is executed locally; non-real-time tasks are directly uploaded to the cloud for processing.
4) Resource Allocation Based on Game Theory. When a large number of terminals request offloading from the same edge node at the same time, it will cause interference to the wireless channel. Based on the non-cooperative game model, each terminal is regarded as a game player. By solving the Nash equilibrium, the system can dynamically adjust the transmission power and computing frequency of each terminal, thereby avoiding "request explosion" on the edge side.

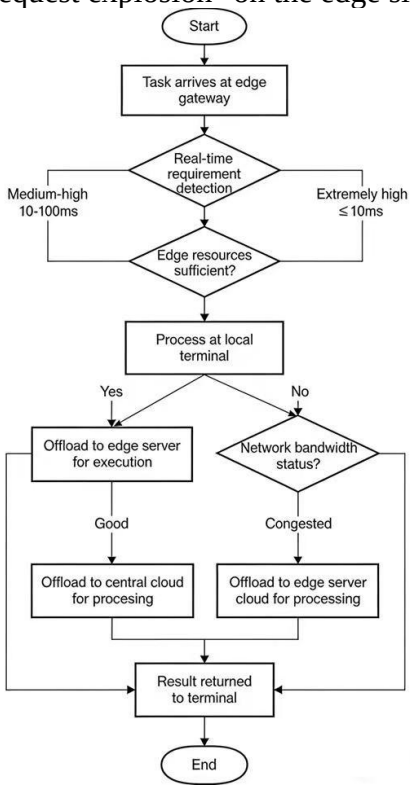

Figure 1. Logical Flow Chart of Computing Offloading Decision

https://doi.org/10.58244/joerai.263732

4 Application Scenarios and Case Empirical Analysis

4.1 Smart City Monitoring and Video Stream Processing

In the traditional "full cloud upload" mode, thousands of 4K high-definition video streams will cause a devastating impact on the municipal backbone network, and the delay of cloud-based identification and alarm often leads to missing the best disposal time.

1) Edge-side local intelligence: Under the cloud-edge collaborative architecture, edge nodes deployed on street poles run lightweight YOLO or SSD detection models. It can perform real-time "pre-judgment and filtering" on video streams: 99% of static backgrounds or normal images are not processed; once traffic accidents, fires or illegal parking behaviors are identified, only a few frames of images containing key features and alarm metadata need to be pushed to the background.

2) Cloud-side global orchestration: After receiving alarm information from hundreds of edge nodes across the city, the cloud conducts cross-regional trajectory tracking and long-term security situation heat map analysis. This mode reduces the bandwidth cost by more than 90% and compresses the emergency response time from the minute level to the second level.

4.2 Industry 4.0 and Flexible Manufacturing Systems

In smart factories, the control instructions of each precision machine tool or robot arm must complete the closed loop in an extremely short time.

1) Edge-side deterministic control: Edge controllers deployed in factory workshops run industrial real-time operating systems (RTOS). They are responsible for collecting high-frequency data such as bearing vibration and temperature, and using edge-side trained models for fault prediction. When a possible tool breakage is detected, the edge side can directly issue a "shutdown" instruction to avoid producing a large number of defective products.

2) Cloud-side lifecycle management: The operation logs of all equipment are regularly uploaded to the cloud big data platform. By comparing the wear curves of equipment in different factories and of different brands, the cloud generates a global preventive maintenance plan. When the cloud discovers a new fault feature pattern through deep learning, it will push the updated model weights to the edge nodes of each factory area, realizing the evolution of swarm intelligence.

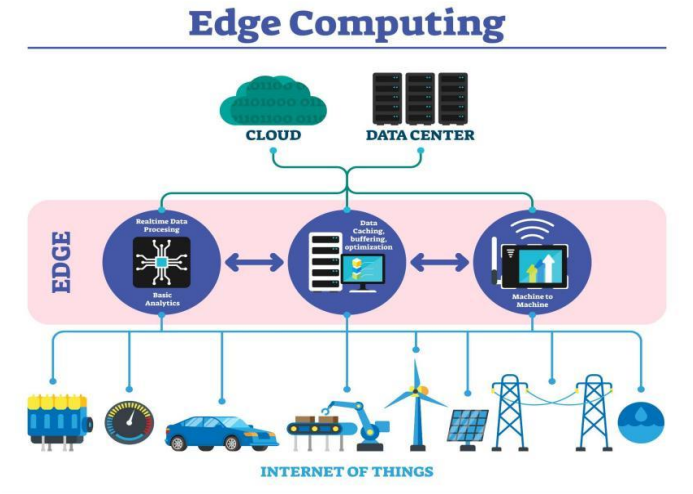

Figure 2. Edge Computing

4.3 Intelligent Connected Vehicle (V2X) Collaboration

In the field of autonomous driving, the perception range of a single vehicle is limited and it is easily affected by obstacles.

1) Edge side: Road side units, as edge nodes, fuse the perception data of lidar and cameras, and broadcast "blind spot warning" or "green wave band guidance" to vehicles within 500 meters in real time. Such ultra-low latency distributed collaboration is the core for the implementation of L4 autonomous driving.

2) Cloud side: Responsible for the dynamic update of high-precision maps and macro traffic flow scheduling to ensure the optimal balance of urban-level traffic efficiency.

https://doi.org/10.58244/joerai.263732

5 Summary and Outlook

From the perspective of distributed systems, this paper systematically studies the evolution, key technologies and applications of the cloud-edge collaborative architecture. Firstly, the evolution context of the distributed system from the master-slave architecture to the cloud-edge collaborative architecture was sorted out. A three-layer logical model of "endpoint device layer - relay edge layer - core cloud computing layer" was constructed, and the functional positioning, characteristics and collaborative mechanism of each layer were expounded. Secondly, the key technologies such as computing offloading decision-making, data consistency maintenance, cloud-edge resource orchestration and communication were deeply analyzed. The offloading decision-making scheme based on game theory and heuristic algorithms, the hierarchical charter consistency mechanism, and the resource orchestration and communication scheme based on KubeEdge and MQTT were proposed. Finally, empirical analyses were conducted in typical scenarios such as smart cities, Industry 4.0, and intelligent connected vehicles to verify the superiority of the cloud-edge collaborative architecture in reducing latency, saving bandwidth, and enhancing system reliability.

Studies show that the cloud side collaborative architecture through "edge processing real-time tasks, the cloud processing global tasks" mode of division of labor, effectively resolve the bandwidth bottleneck of traditional cloud computing architectures and real-time shortage problem, at the same time make up for the defects, calculate the edge of the force limited is the core of the Internet of things era distributed system development direction.

References

[1] Foster,I., &Tuell,S. (2019). Cloud Computing and Distributed Systems: Design and Architecture. Cambridge University Press.

[2] W. Shi, H. Sun, J. Cao, et al. Edge Computing: State of the Art and Future Trends[J]. Journal of Computer Research and Development,2017,54(5):907-924. Zhang,J.,&Liew,S.C.(2020)." Edge-Cloud Computing: A Survey on Architecture and Implementation ". IEEE Communications Surveys & Tutorials, 22(4), 2469-2480.

[3] Buyya, R., &Srirama, S.N.(2018). Fogand Edge Computing: Principles and Paradigms. John Wiley & Sons.

[4] Z. Liu, Research on Resource Scheduling of Distributed Systems in Cloud Computing Environment [D]. Tsinghua University, Beijing, 2021.

https://doi.org/10.58244/joerai.263732