Journal of Computers Vol. 36 No. 3, June 2025, pp. 89-103
https://doi.org/10.63367/19911599202506360300 7

Cloud Computing Network Performance Optimization

Hong-Yao Ju*

Zhejiang Fashion Institute of Technology, Ningbo 315211, China

juhongyao@163.com

Received 19 September 2024; Revised 11 October 2024; Accepted 11 October 2024

Abstract. The core factors determining the cloud computing network (CCN) performance are its structure and working strategy. This study investigated the internal working mechanism of key nodes and collaborative technology between nodes in CCN. Based on the analysis results, we implemented continuous address mapping for several key nodes on the access load processing path, including the load scheduling system, gateway system, application server system, and database system, solving the scalability within key nodes and efficient collaboration between nodes. Applying the read-write separation technology to the load scheduling system solved the problem of path separation between query load and write load in CCN. In the database system, the technologies of mass data segmentation, database publishing, and subscription are integrated, mitigating the problems of selection basis for the quantity of mass data segmentation and low efficiency of mass data processing. This improved the CNN structure and optimized its performance, avoiding several structural and policy design defects of conventional CCN. The performed numerical tests proved that the entire network effectively responded to user access requests even when a single key device failed. Moreover, the developed system exhibited improved performance in high concurrency access and massive data processing.

Keywords: cloud computing network, work strategy, structural improvement, performance optimization, read-write separation, data segmentation

1 Introduction

With the continuous reconstruction of the global industrial structure, the efficient coordination of various links in economic and social activities is imperative, which will further promote the rapid development of economic and social activities to the platform and ecology, such as unmanned driving, expert systems, machine learning, machine vision and so on. Applications related to these economic and social activities require the support of a high-performance CCN system. Therefore, improving and optimizing the performance of the CCN system is an urgent problem that needs to be solved. The high performance of a CCN system requires that its structure combine high collaboration, scalability, and flexibility of application deployment with a strong adaptive ability to support database read-write separation and massive data segmentation processing.

Numerous studies in relevant fields [1-12] have mainly focused on improving or reviewing node algorithms. The introduction of systematic structural improvement and engineering implementation methods for the entire CCN is limited. The CCN system structure and working strategy described in this paper are formed after sorting out the available experience in cloud computing engineering construction in recent years and our relevant invention patent achievements. This paper focuses on the three core parts of the CCN: load scheduling system, application server system, and database system. It describes the CCN system structure improvement method, performance optimization technology, and performance matching strategy among key nodes. The key problems solved in this paper are listed in Table 1.

Table 1. List of the key problems to be solved

Serial number	Problem formulation
1	Multi-gateway redundancy
2	Implementation method of application multi-node deployment
3	The solution of database read-write separation
4	Basic principles for matching database server and application server
5	Query database multi-point migration deployment and synchronization method
6	Relationship between key hardware parameters and data segmentation

* Corresponding Author

2 CCN Structure and Working Principle

2.1 CCN Structure

The CCN system consists of ten parts: external load scheduling system, external network access system, internal load scheduling system, internal network access system, core switching system, application server system, database server system, database synchronization server, database segmentation and merging controller, and data storage system. Fig. 1 shows the specific structure of CCN.

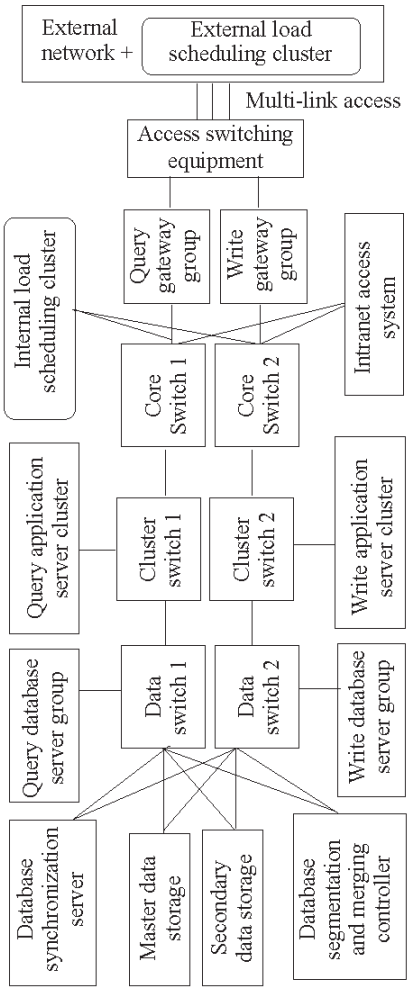

Fig. 1. CCN structure

The external load scheduling system consists of a single load scheduling cluster deployed on the extranet. The external network access system comprises five parts: multiple access links, distribution frame, access switching equipment, query gateway group, and write gateway group. The internal load scheduling system comprises a load scheduling cluster deployed on the Intranet. The internal access system mainly includes the integrated wiring system. The core switching system consists of two core switches. The application server system consists of a query application server cluster, a write application server cluster, and two cluster switches. The database server system consists of a query database server group, a write database server group, a database synchronization server, a database segmentation and merging controller, and two high-performance data switches. The data storage system contains two sets of data stores.

2.2 CCN Working Process

Network users are divided into two categories, including query operators and write operators. When CCN is accessed by Internet or Intranet users, their procedures differ as follows. By accessing the query domain name and then generating the query load, Internet query users get access to the external load scheduling cluster, while Intranet query users get access to the internal load scheduling cluster. Similarly, by accessing the write domain name and then generating the write load, Internet write users get access to the external load scheduling cluster, while Intranet write users get access to the internal load scheduling cluster.

In the case of Intranet users, the query and write loads can be directly dispatched to the query and write application server clusters, respectively.

In the case of Internet users, the query (or write) loads are mapped via the member gateways in the query (or write) gateway group to the query (or write) application server cluster, respectively. These clusters can distribute the query (or write) load to the member server with the highest priority, and the member server can deal with the data required by the application program with the member servers in the query (or write) database server group. Noteworthy is that all member servers in the write database server group use the primary database, while those in the query database server group all use the local subscription databases to provide data services. The subscription database performs segmentation on massive data, providing real-time synchronization with the main database. The specific working process of CCN is shown in Fig. 2.

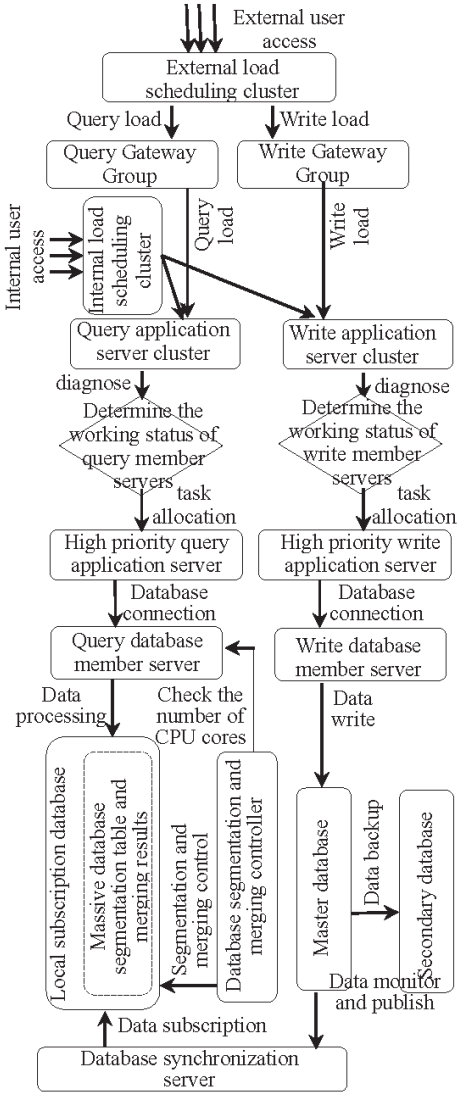

Fig. 2. CCN working process

3 Key Technologies

3.1 External and Internal Load Scheduling Clusters

Load Scheduling Cluster. The load scheduling cluster [13-15] consists of one external and one internal cluster, each with multiple scheduling members. The latter members have a shared cluster VIP address set when the cluster is formed. The load scheduling cluster accepts user service requests through the cluster's VIP address. The structure of the load scheduling cluster is shown in Fig. 3.

In the load scheduling cluster, each scheduling member is configured with a complete set of scheduling records to ensure that each scheduling member can independently complete load scheduling tasks. All scheduling members in the load scheduling cluster work collaboratively under the control of the cluster working strategy and receive automatic monitoring of the working status of the cluster policy. The cluster working strategy assigns work tasks to light load scheduling members with normal working status and high priority.

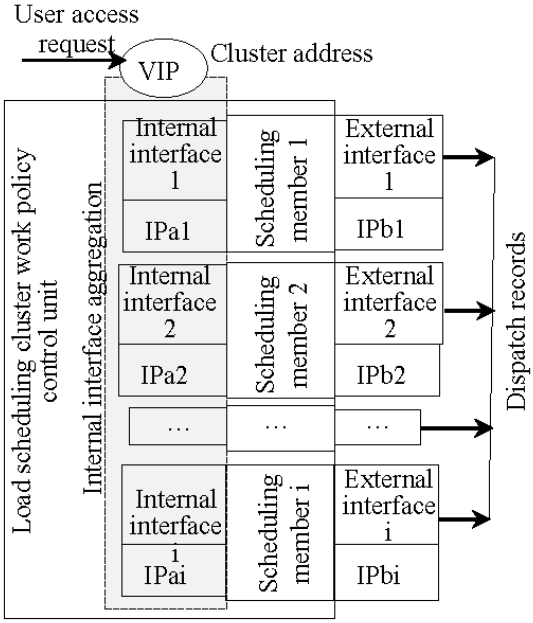

Fig. 3. Load scheduling cluster structure

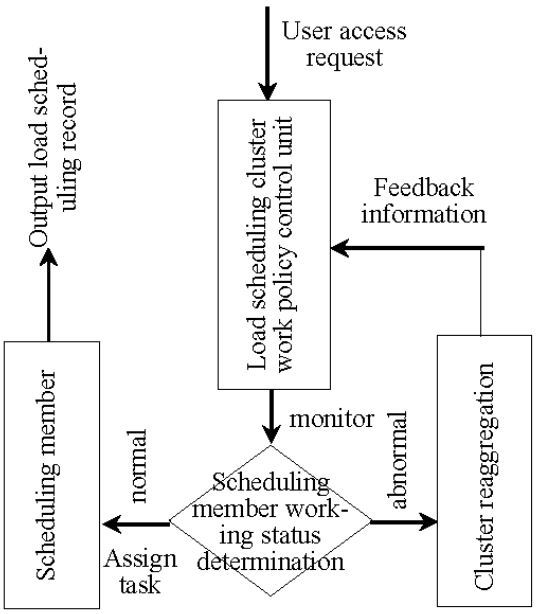

Fig. 4. Load scheduling cluster work policy execution mode

If a scheduling member is abnormal, the cluster policy automatically shields the member, and the task being executed by the member is taken over by the next scheduling member with a higher priority. The cluster policy automatically restores its membership when the abnormal member returns to normal health. The load scheduling cluster work policy is executed as shown in Fig. 4.

Load Scheduling Member. When some extranet users access the network concurrently through a domain name, the domain name is sequentially resolved by the external load scheduling cluster into multiple IP addresses previously bound to multiple access gateways. This IP address mapping method distributes the access load of extranet users to several access gateways. Fig. 5(a) and Fig. 5(b) show the domain name resolution method of scheduling members in the external load scheduling cluster.

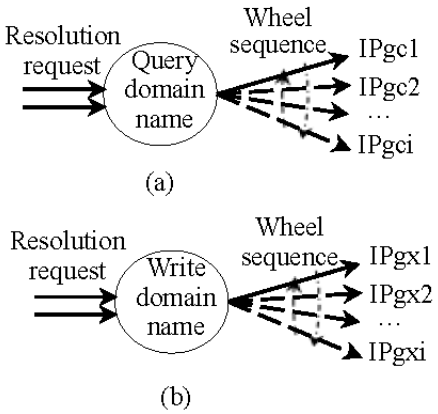

Fig. 5. Scheduling member resolution query (a) and write (b) domain name methods

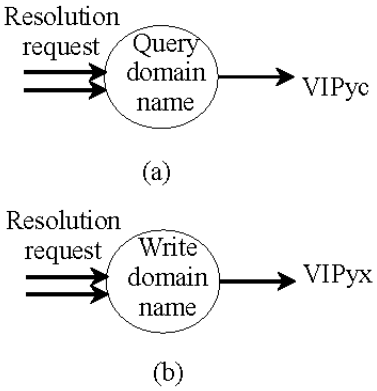

Fig. 6. Scheduling member resolution query (a) and write (b) domain name methods

In Fig. 5, IPgc1, IPgc2, ..., and IPgci are the IP addresses bound to the external ports of the corresponding member gateways in the query gateway group, while IPgx1, IPgx2, ..., and IPgxi are the IP addresses bound to the external ports of the corresponding member gateways in the write gateway group.

When intranet users access the network concurrently using the domain name, the domain name is resolved by the internal load scheduling cluster to the VIP address of the application server cluster. The VIP address is configured when the application server cluster is created. The application server cluster accepts user service requests through this VIP address. The query and write domain name resolution methods of scheduling members in the internal load scheduling cluster are shown in Fig. 6(a) and Fig. 6(b), respectively. Where VIPyc and VIPyx are the IP addresses of the query and write application server clusters, respectively.

3.2 Gateway Groups

The gateway group [16-17] consists of a query and write gateway groups. Each gateway group consists of multiple member gateways, as shown in Fig. 7, where IPg1, IPg2, ..., and IPgi are the IP addresses bound to the external ports of the corresponding member gateways in the query or write gateway group, while IP1, IP2, ..., and IPi are the IP addresses bound to the internal ports of the corresponding member gateways in the query or write gateway group.

The query gateway group accepts scheduling records for query operations from the external load scheduling cluster. The write gateway group accepts scheduling records for write operations from the external load scheduling cluster. During query operations, users use the query domain name to access the network; during write operations, users use write domain names to access the network.

The gateway group's member gateway receives the scheduling record from the external load scheduling cluster according to the IP address bound to its external interface. The scheduling record is the domain name and its corresponding IP address. Each scheduling record is assigned to a specified member gateway based on its IP address.

Therefore, multiple scheduling records resolved from the same domain name by the external load scheduler are evenly distributed among the member gateways of the gateway group in sequence. The goal of load balancing among multiple member gateways in the gateway group is achieved. The collaborative relationship between the gateway group and the external load scheduling cluster is shown in Fig. 8.

The number of member gateways in the gateway group is determined as $Lsum \leq Lg1+Lg2+ \ldots +Lgi$, where *Lsum* is the planned maximum load for the gateway group, *Lg*1, *Lg*2, ..., *and Lgi* are the maximum carrying capacities of the 1st, 2nd, ..., and i-th member gateways, respectively, while i is the number of member gateways.

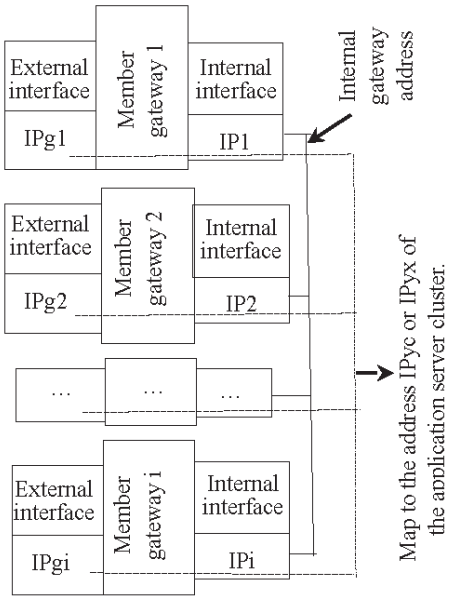

Fig. 7. The structure and working mode of the gateway group

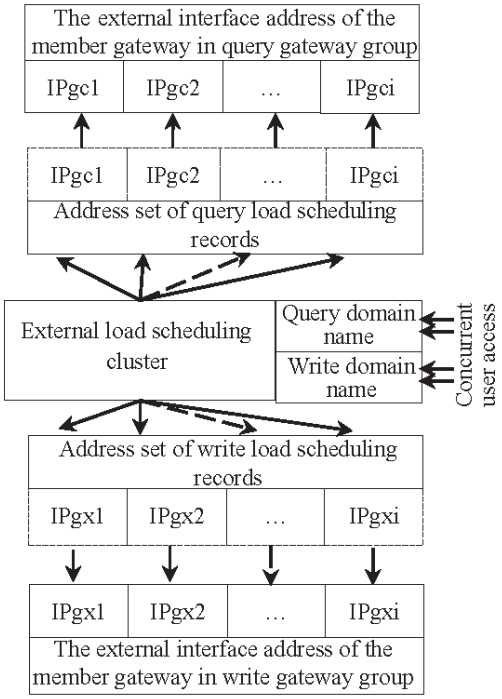

Fig. 8. The collaborative relationship between the gateway group and the external load scheduling cluster

3.3 Application Server Clusters

Structure and Working Process of the Cluster. The application server cluster [18-20] is a load-balancing cluster. This article sets up two application server clusters to achieve the read-write separation function of CCN: the query cluster that executes query tasks and the write cluster that executes write operation tasks. The query cluster accepts query operation tasks assigned by the query gateway group through the VIP address of the cluster and launches the corresponding application for processing. The write cluster accepts write operation tasks assigned by the write gateway group through the VIP address of the cluster and launches the corresponding application for processing.

Each cluster consists of multiple member servers. All member servers in the cluster are aggregated into a whole through their internal interfaces under the control of load-balancing policies. After aggregation, all member servers share a VIP address. The cluster receives access tasks from the outside through the VIP address, and the member server's extranet interface sends the task processing result. The structure and task processing of the cluster are shown in Fig. 9.

In Fig. 9, IPg1, IPg2, ..., and IPgi are the external interface addresses of member gateways in the query gateway group or the write gateway group. VIPy is the address VIPyc of the query cluster or VIPyx of the write cluster.

Cluster Working Policies. The cluster's working strategy [21-23] automatically maps work task packets to member servers through distributed algorithms based on the client's IP address and port number. When the cluster's working strategy receives a work task packet, all member servers simultaneously participate in the mapping decision and select the member server that undertakes the work task. This mapping decision is retained until the cluster structure changes.

The member servers of the cluster coordinate their operations by periodically sending monitoring information. If a member server in the cluster fails or a new member server joins, the working strategy of the cluster triggers

the cluster's aggregation process, updates the status of the cluster, and readjusts the priority of the member servers. During the cluster status update process, the member server undertaking the work task continues to execute the original work task until the work task is completed. The work undertaken by the failed member server is reassigned to the next member server, which is the highest priority in the cluster.

After completing the work task, the member server returns the task result to the cluster policy. The cluster's working strategy changes the source address of the task result packet to the cluster's VIP address and sends the task result to the user.

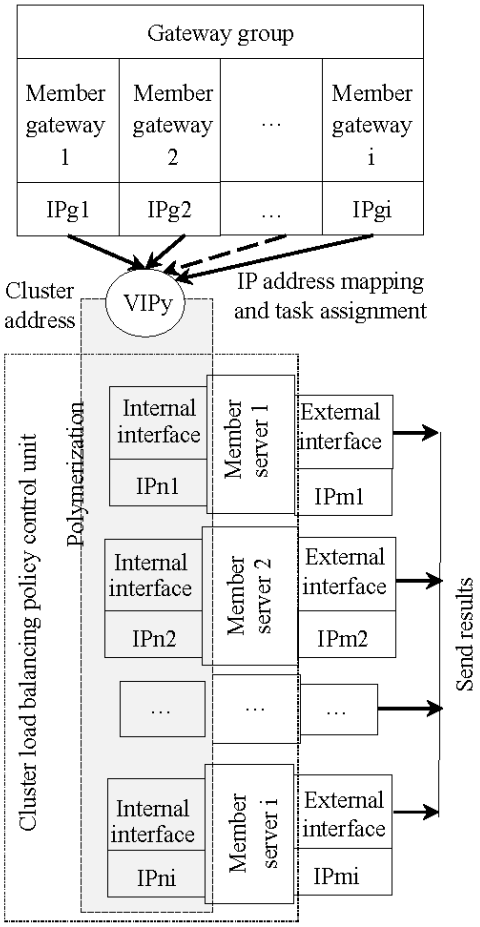

Fig. 9. Cluster structure and task processing

Application Deployment in the Cluster. When deploying an application [24-26] in a cluster, the application must be deployed to all member servers in the cluster, and the deployment mode and deployment environment of the application must be consistent in each member server.

The application's deployment mode is usually published on a website. The website's main parameters are the publication name, application pool, physical path, protocol type, IP address, port number, and hostname. The application's deployment environment usually refers to the application pool the website uses. The application pools provide a proprietary support environment for the applications published on the website.

Fig. 10 shows the application's specific deployment structure in the member server. Projects with * must be the same when the application is published in each member server. The website's IP address must be the cluster's VIP address, not the member server's network interface address.

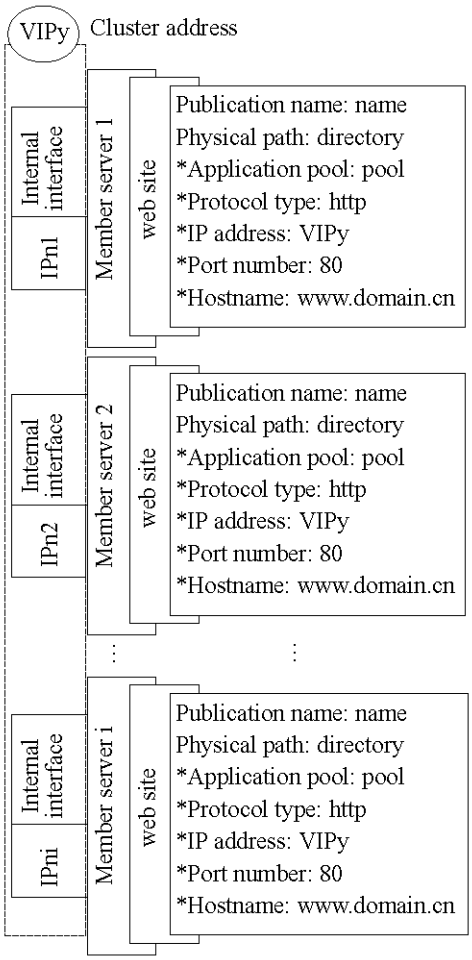

Fig. 10. The deployment structure of the applications in the member servers of the cluster

The hostname used by the website must be the domain name resolved by the load scheduling cluster. When member servers publish applications in the query application server cluster, the hostname used by the website must be the same as the query domain name resolved by the load scheduling cluster. When member servers publish applications in the write application server cluster, the hostname used by the website must be the same as the right domain name resolved by the load scheduling cluster.

If the 64-bit operating system publishes the 32-bit application, the corresponding options in the application pool must be enabled.

3.4 The Query Database Service Units

The query database service unit [27-32] consists of several parts: the query database server group, subscription database, database synchronization server, database segmentation and merging controller, and master database.

Query Database Server Group. The query database server group provides data query services for the query application server clusters. The query database server group consists of multiple member servers, each of which subscribes to the data of the master database and forms a local subscription database. Each member server independently accepts data query requests from member servers in the query application server cluster, and each query request is completed in the member server's local subscription database.

The multi-point deployment method for query databases is convenient for solving the problem of the master database's low response efficiency during high concurrency access. At the same time, data segmentation is per-

formed on tables containing massive amounts of data in the query database. After processing the data table concurrently, the processing result is returned to the member server in the query application server cluster.

Fig. 11 shows the collaborative relationship between the query database server group and the query application server cluster.

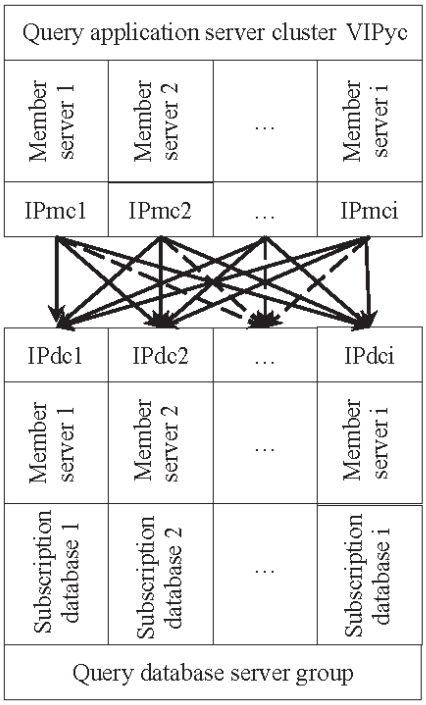

Fig. 11. Collaborative relationship between query database server group and query application server cluster

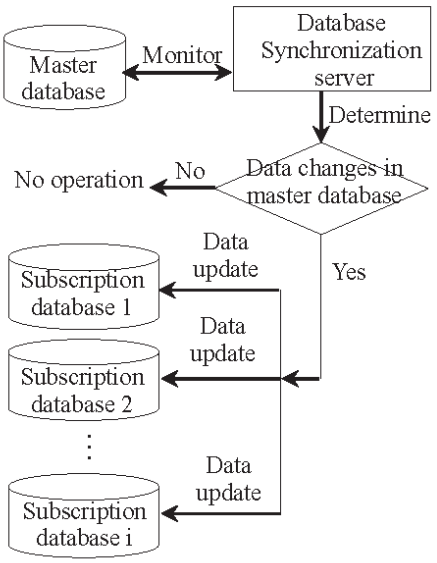

Fig. 12. Synchronization method between subscription and master databases

In Fig. 11, IPmc1, IPmc2, ..., and IPmci are the IP addresses of the external interfaces of the member servers in the query application server cluster, while IPdc1, IPdc2, ..., and IPdci are the IP addresses of the member servers in the query database server group.

Subscription Database. The subscription database is located locally in the member server and provides query services for concurrent tasks. It maintains real-time data synchronization with the master database, which is completed under the control of the database synchronization server. The data synchronization method is shown in Fig. 12.

Database Synchronization Server. The database synchronization server mainly consists of a publisher and a distributor. It undertakes the local subscription database's data synchronization management and control services in the query database server and the master database.

The publisher monitors data changes in the master database in real time and publishes the changed data. After detecting the published data, the distributor updates the subscription database in real time using the published data. The working structure and process of the database synchronization server are shown in Fig. 13.

Database Segmentation and Merging Controller. The database segmentation and merging controller mainly consists of a database detector, a CPU core number detector, a data table cutter, and a result merging device. It is responsible for segmenting large data tables that store massive amounts of data. After the cut data table is processed in parallel, the processing results provide the data query services for the query application servers.

The database detector monitors the subscription databases of the query database member servers in real time. When detecting that the data volume of the data table in the subscription database meets the limit of massive data, the CPU core number detector detects the CPU core number of the database member server. The data table cutter divides the data table into equally sized small data tables based on CPU core values. After parallel processing, these small data tables generate multiple result data tables. All result data is merged into one result data table by the result merging device, providing data services for query requests from the application server.

Through the above data-cutting method, the CPU of the database member server is fully utilized, and the processing efficiency of massive data is significantly improved.

The working structure and working process of the database segmentation and merging controller are shown in Fig. 14.

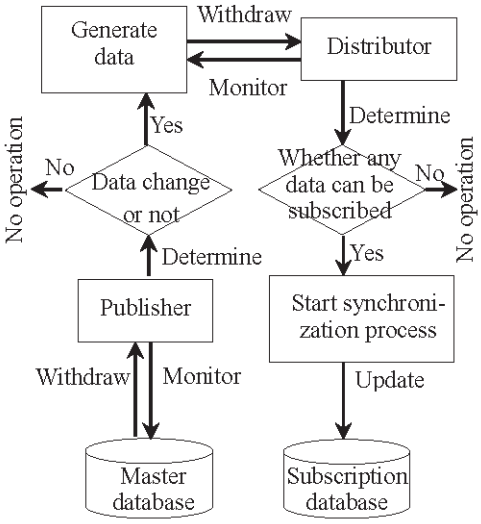

Fig. 13. Working structure and process of database synchronization server

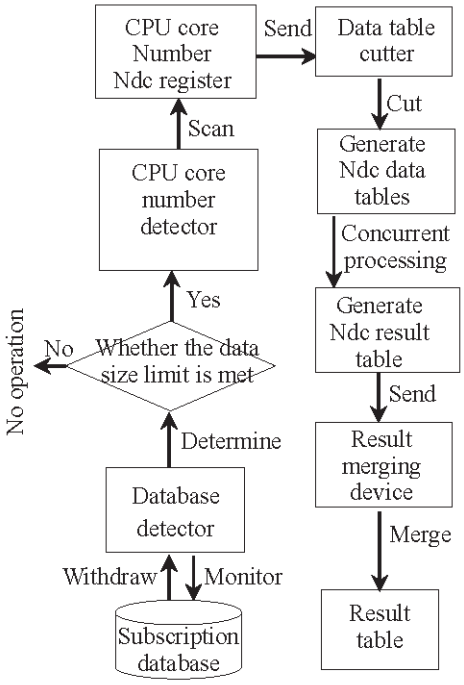

Fig. 14. The working structure and working process of the database segmentation and merging controller

3.5 Write Database Server Groups

The write database server group consists of multiple member servers. Each member server operates directly on the master database in the master data store, providing data write services to the write application server cluster. After receiving the task from the write application server cluster, the member server completes the write operation to the data in the master database through the database management instructions. Fig. 15 shows the collaborative relationship between the write database server group and the write application server cluster.

3.6 Data Storage Units. The data storage unit includes the master and secondary data storage. The master database is placed in the master data storage. The secondary data storage backs up the data in the master data storage in real-time through the network. The working structure of the master data storage and the secondary data storage is shown in Fig. 16.

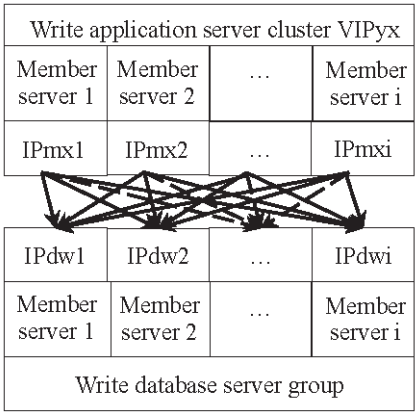

Fig. 15. Collaborative relationship between write database server group and write

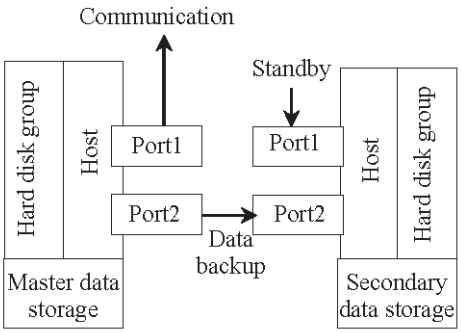

Fig. 16. The working structure of master and secondary data stores application server cluster

4 Test Results and Analysis

4.1 Test Environment

During this experiment, all servers were HP Pro Desk 480, with 8 CPU cores and a Samsung SSD hard drive. The operating system is Windows Server 2022. The database is SQL Server 2014. The basic data table structure of the test database consists of 32 fields and 102400 records.

4.2 Test Results and Analysis

System Response Efficiency. During the testing process, all nodes in the network are dual nodes, and the devices operate normally. We tested the following two situations separately. One situation is the basic data table of the database, and the other is that the basic data table is divided into 8 sub-data tables, where 8 is the CPU core value of the database server. The relationship between the number of concurrent tasks and the system response time value is shown in Fig. 17.

Fig. 17 shows that when the network system is accessed, as the number of concurrent tasks increases, the response time of both the basic data table and the sub-data table continues to increase. This indicates that response time is proportional to the number of concurrent tasks. The response time difference between accessing the basic data table and accessing the sub-data table for each concurrent task node constantly increases. This indicates that the access performance of the sub-data table is higher than that of the base data table.

Fig. 18 shows the relationship between the number of concurrent tasks and the improvement value of system response efficiency after cutting the data table.

The variation relationship of the response efficiency improvement value on the graph indicates that the basic data table is divided into eight sub-data tables according to the number of CPU cores. Since each sub-data table has only 12800 records, the parallel processing of 8 small sub-data tables using an 8-core CPU takes less than 1/8 of the response time of the base data table.

Splitting a data table containing massive amounts of data is one effective method of improving data processing efficiency.

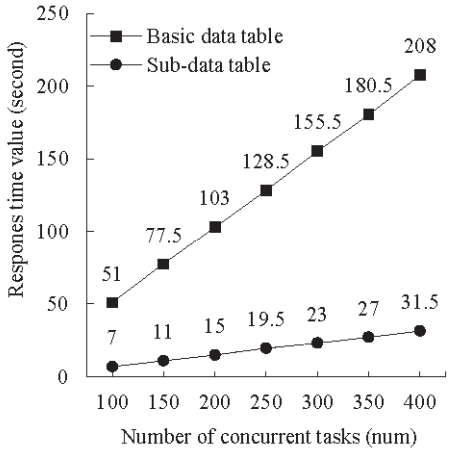

Fig. 17. Relationship between the number of concurrent tasks and response time

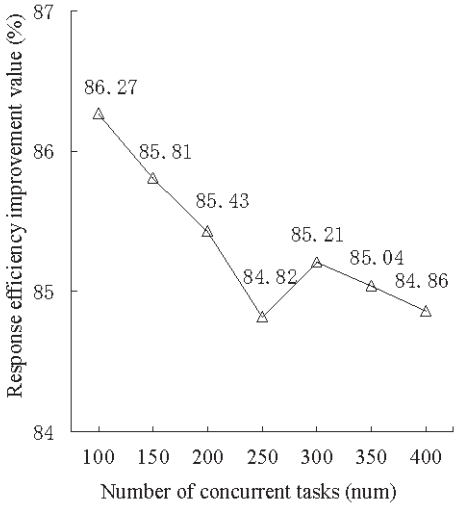

Fig. 18. Relationship between the number of concurrent tasks and response efficiency improvement

Failure Resistance and Scalability of the Application Server Cluster. During the test, we took the query application server cluster as an example and set one query application server as faulty and unable to provide services. The other query application server is running properly.

The test results are as follows. Fig. 19 shows the relationship between the number of concurrent tasks and the value of system response time.

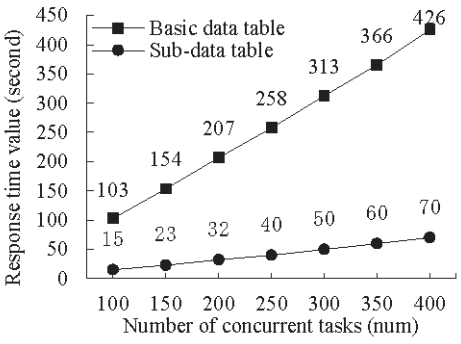

Fig. 19. Relationship between the number of concurrent tasks and response time

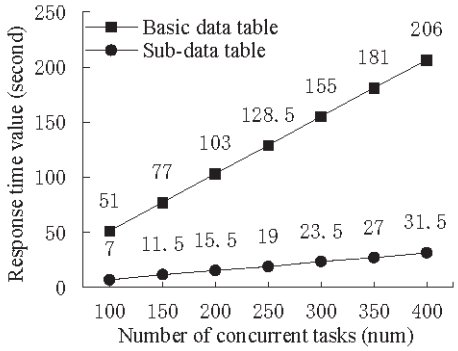

Fig. 20. Relationship between the number of concurrent tasks and response time

As seen in Fig. 19, although one server failed, the whole system can still process user task requests normally, and the system's service function is still normal. However, the system's response time is greatly increased in this case.

The test results show that the cluster system effectively shields faulty servers, and user access requests are scheduled to servers working normally in the cluster. The cluster has good fault resistance and scalability.

Ability of Database Server Group to Resist Failures. During the test, we took the query database server group as an example and set one query database server as faulty and unable to provide services. The other query database server is running properly.

The test results are as follows. Fig. 20 plots the relationship between the number of concurrent tasks and the value of system response time. From the figure, it can be seen that when one database server fails, the whole system can still process user task requests normally, the service function of the system remains normal, and the response time of the whole system has not undergone significant changes.

The test results indicate that the database server group can resist failures. At the same time, they indicate that in this article's experimental environment, the impact of the database server failure on system performance is much smaller than that of the application server failure.

Failure Resistance and Scalability of Load Scheduling Clusters. In the course of the experiment, we conducted the test in two situations. The first case is that a member server in the external load scheduling cluster is set to fail and cannot provide service. The second case is that a member server fails in the internal load scheduling cluster and cannot provide service. The results of the response of the two clusters to the load scheduling record are summarized in Table 2.

Table 2. Test results

Cluster name	Server name	Status setting	Response state
External load scheduling cluster	Member 1	Normal	Normal
	Member 2	Faulty	
	Member 1	Faulty	Normal
	Member 2	Normal	
Internal load scheduling cluster	Member 1	Normal	Normal
	Member 2	Faulty	
	Member 1	Faulty	Normal
	Member 2	Normal	

The test results in the table show that in two load scheduling clusters, as long as a member server is working normally, the cluster can work normally. The test results indicate that the load scheduling cluster has good fault resistance and scalability.

5 Conclusions

The comprehensive analysis of the experimental results obtained made it possible to draw the following conclusions:

1) The multi-level address continuous mapping technology and the load balancing strategy adopted for key nodes, such as the load scheduling systems, access systems, application server systems, and database systems, effectively improve the structure and optimize the performance of the CCN.

2) By using the database multi-node deployment technology and the intelligent segmentation technology for the large database, the role of the high-performance multi-core CPUs in the database server is more efficiently utilized.

3) The efficiency of processing massive amounts of data is significantly improved. At the same time, the CCN system has good failure resistance and scalability.

Acknowledgments

This study was funded by the Pilot Demonstration Project of Manufacturing Industry Entrepreneurship and Innovation Platform of Industry and Information Technology Ministry of The People's Republic of China and the Science and Technology Innovation 2025 Major Special Project of Ningbo City, China (grant no. 2019B10028).

References

[1] Y.-L. Ruan, X.-L. Xu, H. Fa, S.-L. Dong, L. Jiang, L. Yang, Y.-Y. Yan, Research and application of big data platform technology based on multi-centre collaborative computing, Telecommunications Science 40(5)(2024) 141-151.

[2] D.-D. Jiang, Y.-H. Sun, Research on data security issues in big data cloud computing network environment, Communication & information technology (5)(2023) 79-82.
[3] S.-X. Zhao, Z.-P. Zhang, X.-L. Wang, S.-M. Han, L. Yuan, J.-Z. Zhang, Collaborative optimization strategy of edge sensor cloud based on security and low energy consumption, Journal of university of electronic science and technology of china 52(1)(2023) 85-94.
[4] X.-Y. Yang, B.-X. Zhang, Research on cloud computing technology optimization and platform building of network database of big data platform, Journal of yan'an university (natural science edition) 40(3)(2021) 71-73, 78.
[5] W.-X. Han, Research on optimization of computer network security technology based on cloud computing environment, Cyberspace security 14(4)(2023) 93-97.
[6] Z.-Q. Zhen, Practice research on mobile communication network optimization under cloud computing, Construction & design for Engineering (22)(2021) 98-99, 106.
[7] Z.-D. Shan, Y. Zou, Y.-Z. Li, Process optimization and decision-making model of service-oriented manufacturing network based on cloud computing, Computer integrated manufacturing systems 25(12)(2019) 3139-3148.
[8] J. Chen, X.-R. Li, S.-G. Qiu, J. Chen, Technical architecture design for big data storage, Cyber security and data governance 42(s1)(2023) 174-179.
[9] Y. Yan, X. -W. Zhang, L. -J. Wang, Big data analysis based on iterative MapReduce for hybrid cloud, Computer engineering and design 42(4)(2021) 1028-1035.
[10] X. Zheng, B.-L. Du, Research and practice on the infrastructure for operator industry digitalization platform, Telecommunications science 40(5)(2024) 176-185.
[11] C.-X. Wang, J.-Y. Zhang, Y.-L. Niu, Application of low code development platform in enterprise digital transformation, Electronic technology 52(11)(2023) 334-335.
[12] H.-Y. Ju, M. Shi, G.-Y. Lv, Y. Shi, Z.-J. Xie, X.-C. Yao, Z.-Q. Lu, K. Ning, H. Yu, An adaptive measurement and control method and system about concurrent processing massive data, China patent cn202011167948.8, April 11, 2023.
[13] B.-N. Dong, Q.-S. Yang, M.-S. Li, M-RSF: A multilevel feedback queue task scheduling mechanism for unikernel, Journal on communications 45(5)(2024) 54-69.
[14] W.-F. Li, W.-C. Yang, G.-F. Luo, Real-time collaborative scheduling method of production-logistics resources based on swarm consistency, Computer integrated manufacturing systems 28(9)(2022) 2953-2969.
[15] W. Zhang, D. Wang, Real-time demand response scheduling strategy for electric vehicles based on cloud edge collaboration, Power system technology 46(4)(2022) 1447-1458.
[16] C.-Y Liu, C.-B. Si, L. Wang, Y.-H. Wang, Z.-Q. Zhao, Research on optimization path of 5G mobile communication network under cloud computing technology, Changjiang information & communications 37(3)(2024) 209-211.
[17] W.-F Sun, W.-B. Li, Q. Li, Z.-J. Zou, Y. Lu, Fault tolerant communication technology of configuration monitoring system based on integration of DMA and interrupt, Chinese journal of ship research 16(4)(2021) 199-207.
[18] S.-P. Wang, Y. Liang, Design and application of database read-write separation technologies in hospital information system, Techniques of automation and applications 38(6)(2019) 78-80.
[19] Q.-T. Zhang, Research and application of database read-write separation technology based on MyCat, Modern information technology 5(8)(2021) 125-126,129.
[20] X. Zhao, Y. Li, J. Xu, Research of distributed database design using vertical splitting technologies, Control engineering of china 25(1)(2018) 154-159.
[21] H.-K. Omar, K.-H. Jihad, S.-F. Hussein, Comparative analysis of the essential CPU scheduling algorithms, Bulletin of electrical engineering and informatics 10(5)(2021) 2742-2750.
[22] B. Zheng, Design of enterprise container cloud platform based on kubernetes, Digital technology & application 37(6)(2019) 138, 141.
[23] F.-L. Chen, Experience sharing of edge cloud construction for telecom operators, Communication & information technology (5)(2024) 69-72, 114.
[24] X.-L. Hou, Research and implementation of multi-dimensional space reverse nearest neighbor query algorithm, Computer & digital engineering 46(4)(2018) 672-676.
[25] G. Lu, C.-Y. Chen, Z.-L Huang, Z.-Y. Huang, Research on intelligent cloud native architecture and key technologies for cloud and network integration, Telecommunications science 36(9)(2020) 67-74.
[26] J.-F. Yan, J.-M. Tang, Short-term demand task scheduling optimization algorithm for large-scale cloud computing network users, Modern electronics technique 47(6)(2024) 63-67.
[27] Y. Sun, S.-M. Huang, Z.-K. Li, B. Li, M.-L. Mu, X.-D. Zhang, F. Li, Joint control strategy of heterogeneous temperature control load considering time domain characteristics, Power system technology 44(12)(2020) 4722-4734.
[28] Y. Xu, The application of artificial intelligence in computer network technology in the era of big data, Video engineering 47(4)(2023) 142-144.
[29] B. Sheng, Analysis and design of cloud computing platform network architecture based on OpenStack, Electronic design engineering 27(15)(2019) 45-50.
[30] Y. Cai, R. Fan, S. Jin, Research on key technologies and networking strategies for lossless networks in data centers, Designing techniques of posts and telecommunications (7)(2024) 83-87.
[31] C.-Z. Zhao, Z.-P. Cai, Z.-X. Hong, Cloud-edge-based intelligent production line multi-objective task scheduling, Electronics quality (11)(2023) 33-38.

[32] X.-M. Xiong, J. Zhao, Digital research and implementation of cloud-network planning of Telecom operators, Telecommunications science 38(11)(2022) 163-168.