Research Article

Research on Load Balancing and Caching Strategy for Central Network

Zhi-hong Wang and A-liang Dong

Hebei Academy of Fine Arts, Xinle, Hebei 050700, China

Correspondence should be addressed to Zhi-hong Wang; wzh@hbafa.edu.cn

Received 16 February 2022; Revised 7 March 2022; Accepted 18 March 2022; Published 8 April 2022

Copyright © 2022 Zhi-hong Wang and A-liang Dong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The central network architecture takes the information content as the processing object and can better meet the user access needs. However, the traditional network node load balancing and caching strategy has been difficult to adapt to the development of the central network architecture. Therefore, this article proposed a load balancing and cache management strategy based on the central network in order to provide a reference for the architecture and application of the central network. Based on the analysis of the existing network architecture, this article expounded the networking form and characteristics of the central network system. By analyzing the load scheduling requirements of network nodes, combined with the transmission characteristics of resource requests in the central network, a cache node load balancing method suitable for the requirements of the central network was proposed based on the existing load balancing algorithms. In addition, by analyzing the cache management strategy and its impact on the performance of the network system, a cache node information forwarding model with resource requests as the object in the central network environment was given. Finally, through experimental test and comparative analysis, the results showed that the method proposed in this article had certain feasibility and effectiveness. Compared with other typical strategies, the load balancing and caching strategy proposed in this article can better meet the user needs of the central network.

1. Introduction

With the expansion of the Internet application scale and the continuous growth of the number of network users, the way of information acquisition has changed from early resource sharing to information resource distribution. The traditional communication mode based on TCP/IP protocol has been greatly limited. In order to meet users’ access requests to the network, some researchers have proposed some new network structures based on the traditional end-to-end network structure, for example, CDN, P2P, web caching, and other network architectures. However, because the traditional TCP/IP protocol is still used at the bottom of these network architectures, it does not really solve the problems of Internet application expansion and increasing user demand [1, 2]. With the deepening of mobile Internet and its application, a new form of central network architecture has been introduced in recent years. Driven by the receiver, the network has good caching capacity and forwarding information based on content and has the characteristics of node mobility to a certain extent.

The central network mainly forwards the content requested by the user as the center and accesses and searches the requested content in the whole network, regardless of the specific storage location of the content, to realize the separation of the accessed content and the specific location. The information forwarding mode of the central network has evolved from the original storage medium to cache forwarding and solves the problem of massive information transmission congestion or limited rate. In recent years, some scholars have proposed a variety of ICN architectures, for example, the more mature fully distributed NDN (named data network) network architecture in applications [3, 4]. Because different forms of central network architecture are complex in networking design, some scholars have studied how to ensure the efficiency, security, and reliability of the
content requested by users in the process of distribution, which has also become one of the research hotspots in the field of the central network.

Because the central network forwards the information with the request content as the object, the central network has the characteristics of responding to the information service and responding to the nonconnection request. When the user requests the content, the network receiving node does not need to find the specific location of the server but matches the relevant string in the whole network by requesting the content in order to find the node containing the content. This way of searching through content can not only meet the requirements of users but also meet the needs of network development.

Because the central network is a network operation form centered on user request content, limited by network technology and its conditions, the main factors affecting the development of the central network are network node load scheduling and cache management. Therefore, starting from the networking structure characteristics of the central network, this article analyzes the network node load scheduling process, expounds the network node balancing processing method, and puts forward the load balancing and cache management strategy for the central network according to the cache demand characteristics, to provide reference and support for better realizing the network user request.

2. Related Works

In recent years, scholars at home and abroad have carried out a lot of related work on network load balancing and achieved some theoretical and practical results. By comparing different load balancing strategies in the cloud environment, some scholars know that the static load balancing method is simple and easy, but it is difficult to adapt to the dynamic changes of dense network nodes. The dynamic load balancing method can better adapt to the distributed heterogeneous cloud environment, although the calculation is complex and the time cost is large [4, 5]. As a common static load balancing method, the weighted polling strategy mainly distinguishes the differences between service nodes by setting different weights. Among them, the service node with a larger weight will be allocated to more user requests through polling, while the service node with a smaller weight will get fewer user requests.

The dynamic load balancing method mainly adopts increasing complexity to deal with the constantly updated network environment. In order to improve the utilization of request resources, some researchers propose a load balancing strategy based on a genetic algorithm to adapt to the changing network environment and request information. The performance parameters of network service nodes are polled according to a certain period, the real-time load of each service node is calculated, and whether to provide requested services for users is determined according to the load parameter value to ensure the effective implementation of load balancing strategy. In addition, the load balancing strategy has been gradually applied to related products. For example, the Network Dispatcher developed by IBM can calculate the number of connections of each node in the network in real time and balance the load distribution according to the number of connections owned by each node [6, 7]. F5 Networks designs and develops an F5-BIG-LTM load balancer by comprehensively using static and dynamic load balancing methods, which can track the real-time performance of each node of the network, to effectively optimize the resource allocation [8]. Although the load balancing method has been applied in the server cluster, it is still in the development stage in the application field of the central network.

Network caching technology is very important to solve the shortcomings of network software and hardware technology. Cache design can not only reduce the occupation of network bandwidth but also reduce the time overhead of user requesting content and reduce the load of network nodes to a great extent [9]. Compared with the traditional TCP/IP network, different caching technologies have been widely used in the central network environment. The cache technology for the central network mainly focuses on two aspects. On the one hand, it is about the design and implementation of the cache operation mechanism. Through specific caching technology solutions to improve the operation performance of the central network, including the distribution of cache resources, the replacement method of cache content, and the caching method of requesting content. When designing the caching mechanism, the neighborhood cache is mainly used to solve the problems of excessive node load and information congestion caused by centralized user requests, which can enhance the service provision ability and robustness of the central network nodes. Some key problems need to be solved in the design of caching mechanism. For example, periodically finding content in the cache may increase the cache replacement time overhead, and caching without requesting content will waste cache resources. In addition, when the data packet is returned, if the data packet is backed up on each cache node, it will also cause cache redundancy.

The other is the modeling and analysis of the cache network. By simplifying and abstracting the central network, the corresponding network model is established and its performance is analyzed [10]. The modeling and analysis of the cache network are very important for the operation of central network. The research on network modeling is generally based on linear or tree methods to analyze the performance parameters such as cache hit rate of network nodes, response time of user requests, and number of node hops. In addition, the wireless central network architecture based on the combination of WLAN and central network can better meet the development needs of the times, and the corresponding network cache node load balancing processing method has become one of the issues closely concerned by scholars in related fields.

3. Distribution and Characteristics of the Central Network System

3.1. Common Network Models. With the improvement of communication and computer level, the Internet has developed rapidly, and the bandwidth, software, and hardware
of network broadband are constantly upgraded. In order to meet the increasing needs of users for network use, the organization and architecture of network system are also changing. These networks mainly include CDN (content delivery network) system, P2P (peer-to-peer) network, and CN (central network) [11, 12].

CDN is a network architecture based on content distribution proposed by the research team of the Massachusetts Institute of Technology in 1996. The purpose is to build a content distribution platform through the Internet to provide relevant services for different websites in order to improve service quality. Usually, a simple CDN network mainly includes a DNS server and several cache servers mainly adopt different types of cache servers and distribute them in areas where users’ access is concentrated. When a user requests access to the website, the CDN will transfer the user's access to the nearest cache server in order to quickly respond to the user's requests' system can avoid various factors affecting data transmission on the Internet and ensure that relevant information can be transmitted quickly and accurately. Because the intelligent virtual network can be built by setting up multiple node servers in different areas of the network, the CDN system comprehensively considers the load of each node of the network, the distance from the node to the user and the response time, and transfers the user's access request to the network node nearest to the user. By responding to user access requests closely, the purpose of alleviating Internet congestion can be achieved when the system alleviates the traffic of the backbone network and improves the related service quality to a certain extent. Because the CDN system adopts the networking service mode of Client/Server, in order to meet many information transmissions needs, many CDN servers must be added, increasing the network maintenance cost and limiting its expansibility. The schematic diagram of CDN networking architecture is shown in Figure 1.

P2P system belongs to overlay network, which is a way of network information exchange. Each user of a P2P network is considered a peer-to-peer node, and the resources are shared equally by each node. When a pair of nodes are connected, they can exchange resources with each other. Each node can obtain the required resources from other nodes as a client, and the node can also provide the resources to other nodes as a server. Different from the previous client/server networking methods, the performance of a P2P network is not determined by the number of servers but depends on the performance of peer nodes and related technologies. Due to the resource sharing characteristics between peers in the P2P network, when the number of peers increases, the performance of the P2P network also increases. Therefore, compared with the CDN system, the P2P network has better scalability and can adapt to large-scale networking. Because P2P network nodes can provide client and server functions to other peers in the network, P2P can not only effectively solve the problem of affecting the quality of service due to too busy servers in C/S networking mode but also solve the problem of load imbalance caused by resource scheduling. The schematic diagram of P2P networking architecture is shown in Figure 2.

Although the previous networking methods can improve the network service quality to a certain extent, with the emergence of mobile Internet and the significant growth of the number of connections, the existing networking methods have been difficult to meet the growing access needs of mobile devices. Therefore, with the wide application of the 5G network, setting multiple low-power base stations in the central network can not only improve network coverage and increase network access but also reduce access delay and power consumption. By setting up multiple small base stations in the central network, not only is the capacity of the network increased, but also the performance of the network is greatly improved. In the central network, the intensive deployment of base stations not only shortens the transmission distance between nodes but also significantly reduces the network delay. In addition, the intensive deployment of small base stations in the central network can not only facilitate the rapid switching of users in the central network but also effectively reduce the workload and energy consumption of the macro base station. The schematic diagram of the Dense Central Network (DCN) networking architecture is shown in Figure 3.

3.2. Architecture and Characteristics of the Central Network System. The architecture of the central network generally adopts two ways: one is the networking mode of the macro base station and micro base station, and the other is the networking mode of the micro base station and micro base station. In the first mode, the macro base station is used to transmit services requiring low transmission rate but high mobility, while the micro base station is used to transmit high bandwidth services. The macro base station is mainly used to ensure that the network has a certain coverage and carry out collaborative management of micro base stations. According to the business development needs and user distribution characteristics, multiple micro base stations can be flexibly deployed centered on the macro base station to meet the network needs of different users. This networking method can not only optimize the network coverage area and capacity but also solve the switching problem of user access needs between different areas in the central network environment to truly meet the user's demand experience and improve the utilization of resources.

In the second networking mode, since the macro base station is not adopted, in order to realize the resource coordination function of the first networking in this mode, a dense network composed of multiple micro base stations is adopted as the virtual macro base station. Virtual macro base station shares signal, carrier, channel, and other resources through multiple micro base stations. Since each micro base station sharing the same resources can realize the rapid transmission of user demand information, the purpose of resource management of virtual macro base station can be realized. In addition, because each micro base station can transmit user data independently, it can meet the network access needs of different users in the central network environment.
Compared with the traditional communication network, the central network has the following main characteristics [12]:

1. Many micro base stations are deployed around users. Each access point of the central network is a low-power micro base station with small coverage, and the distance between each access point is usually within tens of meters. The coverage area of the central network is different from the traditional communication network, in which most users are close to the micro base station.

2. The micro base stations around the user interfere with each other. Due to the close distance between the access points of the central network, there is some interference between the micro base stations, which has a certain impact on the performance of the central network.

3. The number of users involved in the central network is huge, and the deployment of micro base stations is dense. Generally, it is difficult to ensure that each micro base station can normally complete the backhaul transmission. If it cannot meet the backhaul transmission requirements of high transmission rate and low transmission delay, it may affect the data exchange between different nodes in the transmission process, which is also the main factor affecting the performance of the central network.

With the increase in the number of central network nodes, the coverage areas between each node overlap each other, resulting in mutual interference. Due to certain interference between different network areas, users’ access to the network is frequently switched, which increases the resource overhead in the process of data transmission. In
addition, because many access points are deployed around the central network, the load balancing and fast backhaul transmission of access points are also the main problems that the central network needs to face.

4. Load Balancing Method

4.1. Load Balancing Strategy. At present, the common load balancing methods mainly include polling, weighted polling, and minimum number of connections. These methods can meet the load balancing to some extent, but the cache hit rate is not high, so they are not suitable for the networking mode facing the central network. From the existing research on the application of load balancing methods, Hash Load (HL) method, as a load balancing strategy, can be better applied to dense network environment [7]. It can not only ensure good load balancing but also have a high cache hit rate, which is suitable for the central network.

When the hash method is used to distribute the user request to the network node, the hash key requested by the user and the hash value of each node in the system need to be calculated by the hash function, and then the node with the largest hash value is used for the service. When the central network needs to add or delete nodes, the hash method has little impact on system performance and load migration. However, when using this method, the load balance of the system is related to the randomness of the hash function, and it takes a lot of time to calculate the hash values of user requests and nodes.

In order to improve the shortcomings of the hash method, in recent years, some scholars have used the hash function to output the relationship between the service node and the user request on the system to the hash ring, which is called Consistent Hashing Load (CHL) [8]. Starting from the position of the user request on the hash ring, the first node searched in a clockwise direction is used as the user service node. This method reduces the time complexity of the hash strategy in calculating the hash values of all nodes. By calculating the hash values requested by client users, we can quickly find the nodes providing services on the hash ring. The working diagram of this method is shown in Figure 4.

As can be seen from Figure 4, when the above hash method is adopted, the load balance of the system depends on the hash values of two adjacent nodes on the hash ring. When the nodes are evenly distributed on the hash ring, the load balance of the system is good. However, if the distance between two nodes on the hash ring is much greater than that between other nodes, the system load will be unbalanced.

4.2. Improved Load Balancing Method. Because a resource in the system is more popular with users than other resources, the same user requests may be centrally allocated to a node, resulting in hot node problems, which will affect the load balance of the system. In order to avoid the centralized allocation of request resources to a node, you can set a load
upper limit for each node. This is the common Consistent Hashing with Bounded Loads (CHWBL) method [13, 14].

At a certain time \( t \), when the user requests to pass through the load scheduler, the load scheduler first calculates the number of task connections of each downstream node and takes it as the load of the node. Then, the load scheduler calculates the total load \( L_t \) of the system at this time, which is shown as follows:

\[
L_t = \sum_{i=1}^{n} L_{st} + L_{\text{new}},
\]

where \( L_{st} \) represents the number of connections owned by network node \( s \) at time \( t \), \( L_{\text{new}} \) is the number of new connections received by the load scheduler, and \( n \) denotes the total number of network nodes.

By calculating the ratio of the total load of the system to the number of nodes, the average load \( L_{\text{avet}} \) of the system at time \( t \) can be obtained. The calculation formula is as follows:

\[
L_{\text{avet}} = \frac{L_t}{n}.
\]

In order to ensure that each node in the system can provide services normally, the upper load limit of each cache node is usually set to 1.2 times the average load. When the user requests allocation, the load scheduler will determine the direction of request allocation according to the load of each node. When the node load has reached the upper load limit, the node will no longer receive user requests from the load scheduler so that the load borne by each node is close to the average load to maintain the load balance of the system.

Although setting the upper load limit of a node can solve the problem that resources are too concentrated on a node, the load balance of the system is still affected by the distribution of each node on the hash ring to a certain extent. For example, there are 5 different nodes distributed on the hash ring, as shown in Figure 5.

![Figure 4: Cache node location relationship on the hash ring under CHL policy.](image)

Suppose \( a_1 \) to \( a_5 \) successively represent the difference of hash values of two adjacent different nodes on the hash ring, where \( a_1 \) is the largest and \( a_5 \) is the smallest. When each node does not reach the upper load limit, the user request corresponding to \( a_1 \) area will be allocated to node \( S_1 \) to provide services, while the user request corresponding to \( a_5 \) area will be allocated to node \( S_5 \) to provide services, and so on. The probability \( P_i \) of the user requesting to obtain the service of a node on the hash ring has a uniform distribution law, as shown in the following formula:

\[
P_i = \frac{a_i}{\sum_{i=1}^{n} a_i}.
\]

Since \( a_1 \) to \( a_4 \) are much larger than \( a_5 \), requests from users are most likely to be allocated to nodes other than \( S_5 \) for processing. Although setting the upper load limit value of each node can ensure that the real-time load of each node does not exceed the upper load limit, in the dense central network environment, when a node processes the relevant requests, it can accept new user requests, which will lead to the overload of some nodes on the hash ring, thus affecting the load balance of the system. For example, according to the
distribution of each node on the hash ring in Figure 5, according to the probability calculation formula of each node processing the request, it can be known that the new user request will be assigned to node S1 for processing. When the real-time load of node S1 exceeds its upper load limit, the user request will be transferred to node S2 for processing, and so on. Therefore, when node S2 to node S4 cannot receive the user request transferred by node S1 because they exceed the load upper limit value, only node S5 will receive the user request at last. If node S1 can receive a new user request after processing the existing request, the probability of node S5 obtaining the user request will be very small.

In order to solve the problem of uneven distribution of nodes in the hash ring, several virtual nodes with mapping relationships with real nodes can be created on the hash ring [15]. Therefore, each real node may correspond to several virtual nodes. When the number of real nodes and virtual nodes reaches a certain number, the nodes on the hash ring will show a trend of uniform distribution. When the user requests resource allocation, the user request is first allocated to the virtual node. At this time, the corresponding real node will provide actual services for the user request. The load balance of the system can be guaranteed by introducing a large number of virtual nodes. The positional relationship between real nodes and virtual nodes in a hash ring is shown in Figure 6.

In Figure 6(a), there are four nodes, S1, S2, S3, and S4, on the hash ring. Due to the small number of nodes, the hash values between adjacent nodes vary greatly. According to the probability calculation formula, the number of user requests obtained by S4 is greater than that of the other three nodes. In Figure 6(b), each node corresponds to multiple different virtual nodes, and the virtual nodes are randomly distributed on the hash ring so that the hash ring is further subdivided by multiple nodes. The greater the number of nodes, the smaller the hash value difference between adjacent nodes. When the number of nodes reaches a certain value, the more evenly the nodes on the hash ring are distributed, the better the load balance of each node in the network is.

5. Cache Management Policy

5.1. Cache Management Strategy and Its Impact on System Performance. Cache technology is the key to improving the performance of the network system, and cache strategy is the core of cache technology. Due to the limitation of cache space capacity, when the cache storage space reaches the upper limit, the objects applying for the cache cannot enter the cache space. At this time, an effective cache strategy needs to be adopted to move the objects that do not need to occupy the cache space out of the cache to free up enough space for urgent objects. The quality of cache strategy has a great impact on the hit rate of cache nodes and the response time of the system. At present, common cache management strategies mainly include LRU (Least Recently Used), LFU (Least Frequently Used), SIZE, and LRV (Lowest Relative Value) [16, 17], which are shown in Figure 7.

LRU is a typical cache policy based on access time interval, which is shown in Figure 7(a). The LRU places the recently accessed cache object at the head of the cache queue. When the cache needs to free space, it will move out the cache object that has not been used for the longest time at the end of the queue. Although the algorithm of LRU is simple, some cache objects that need to be accessed again may be deleted because only the latest access time is considered in application.

LFU is a cache policy based on the number of accesses, which is shown in Figure 7(b). When the cache needs to delete objects, the LFU moves the object with the least number of accesses out of the cache space. When there are objects that have been accessed many times but have not been accessed for a long time, this method wastes cache space to a certain extent.

SIZE is a cache management policy based on the document size, which is shown in Figure 7(c). SIZE receives more small document objects by removing large document objects from the cache. Although it improves the cache hit rate to a certain extent, it increases the cost of loading large document objects. When the SIZE cache management policy is applied, if the large cache object is removed and accessed again, it may produce more access delay and a lower cache hit rate.

LRV is a cache management strategy based on retained value, which is shown in Figure 7(d). By comparing and analyzing the value of cache objects, LRV moves out cache objects with low retention value and large impact on cache hit rate. Compared with other cache management strategies, when calculating the value of different cache objects, LRV fully considers the parameters such as object access times, document size, and recent access time, so it can calculate the retention value of cache objects relatively accurately. Because this strategy needs to comprehensively consider the retention value of different cache objects from many aspects, it may affect the user’s access delay to the network due to the long calculation time.

5.2. Cache Management Strategy Design. When a user makes a request, the retrieval and distribution of the requested content in the central network are mainly completed
through interest packets and data packets. The interest package contains the retrieved content and is used to request the data package of the required content. The content receiver forwards the interest packet to the idle port in order to send the requested content. When the node in the network receives the interest packet, if the node is the publisher of the content or contains the corresponding data packet in the cache, the data packet is returned from the port where the interest packet arrives so as to forward the data content to the requesting user. Nodes in the network need to constantly update the content cache (CS), pending interest table (PIT), and forwarding information table (FIB) [18, 19].

In the central network, users and service nodes interact with each other through interest packets and data packets. After the user sends the interest packet, the network node will respond to it according to the load scheduling and cache management policy and return the request content to the user through each node. The node load scheduling and its cache management model are shown in Figure 8.

When the user makes a network request, first look up the FIB table and send the interest packet from the port where the content can be obtained according to the FIB table item. When the interest packet arrives at the corresponding node, first find the requested content in the cache. If it hits, copy
the corresponding data from the cache, construct a data packet, and forward it from the arrival port of the interest packet. If it is missed, it will search in the PIT table. If the requested content is found in the PIT table item, it will add the interest package arrival port in the PIT table item. If the request content is not found in the PIT table item, first add a new table item recording the request content in the PIT table, then look up the FIB table, and forward it from the recording port in the FIB table. When a node containing the same request content receives an interest packet, it encapsulates the corresponding content in the packet and forwards it from the interest packet arrival port. When the packet arrives at the node, first put the contents of the packet into the cache and then look up the PIT table. According to the feedback content table item recorded in the PIT table, forward the data packet from the corresponding port and delete the corresponding table item in the PIT. The data packet is finally fed back to the user, and the user obtains the required content from the data packet so as to complete a data request. If the user request times out and does not receive the packet, the same interest packet can be sent again.

6. Experiment and Analysis

6.1. Load Balancing and Cache Policy Evaluation Index. In order to quantify the load balance in the central network environment, the load balance degree $B$ is usually used for evaluation [7, 8]. Load balance refers to the ratio of the sum of the absolute value of the difference between the actual load and the theoretical load of each node of the network to the actual total load of the network. The smaller the load balancing degree, the better the load balancing performance of the network. If the actual load of the network node $S_i$ is $L_{PSI}$ and the theoretical load is $L_{TSI}$, the calculation formula of load balance degree $B$ is as follows:

$$B = \frac{\sum_{i=1}^{n} |L_{PSI} - L_{TSI}|}{\sum_{i=1}^{n} L_{PSI}},$$

where $n$ represents the number of network nodes and $S_i$ represents the $i$ th network node. In the central network environment, $L_{TSI}$ represents the average theoretical load of the network.

When the user requests to reach the network cache node, if there is the content requested by the user in the node, it is called a cache hit [20]. Therefore, the cache hit rate $H$ can be used to represent the effect of cache policy execution in the central network environment, and its calculation formula is as follows:

$$H = \frac{N_y}{N_a} \times 100\%,$$

where $H$ is the number of hits of the network cache node and $N_a$ is the total number of requests of the user.

From the time when the client sends a user request to the network load scheduling equalizer to the time when the client receives the result returned by the load scheduling equalizer, this time interval is called the response time of the central network [21]. The response time of the central network can be obtained by relevant special performance test software.

In the central network environment, the amount of data sent from the server sending port is generally much larger than that received by the receiving port. In order to evaluate the traffic loss in the process of data reception and transmission, the network traffic consumption $C_t$ can be described [8], and its calculation formula is as follows:

$$C_t = \int_{t_{start}}^{t_{end}} (v_t \times t) dt,$$

where $v_t$ represents the real-time transmission rate of data sent from the server transmission port, $t$ represents the time taken for data transmission, $t_{start}$ represents the time when the data transmission starts, and $t_{end}$ represents the time when the data transmission ends.

The number of network node connections mainly refers to the number of requests contained in each node of the central network. Generally, the number of network nodes can be used to represent the actual load of network nodes, and it can also be used to evaluate the congestion of network nodes.
6.2. Results and Analysis. In the experiment, the load balancing degree is used to test the effect of the load balancing strategy proposed in this article on the network performance, and several typical load balancing algorithms are selected for comparison in order to verify the superiority of the load balancing method proposed in this article. At the same time, the load distribution of network nodes is tested by using this method, as shown in Table 1.

The load balancing strategy of this article is tested in the concurrent environment with network loads of 2000, 4000, 8000, 10000, and 12000, respectively, and the load distribution of relevant nodes in the central network environment is obtained as shown in Figure 9.

In the concurrent environment with a network load of 8000, six polling requests are made for resources. Combined with the cache hit rate calculation formula, the cache node hit rate under different load balancing strategies in the central network environment is obtained without considering the upper limit of cache node load, as shown in Figure 10.

In addition, combined with the request response time calculation formula, the comparison of service response time under different policies in the central network environment can be obtained, as shown in Figure 11.

From the above experimental comparison results, the load balancing and cache management method proposed in
this article can be better applied in the central network environment. Especially, it is superior to other typical methods in the evaluation indexes such as load balance, cache hit rate, and node response time.

7. Conclusion

For the central network, the network node cache technology can reduce the network backhaul load, reduce the access delay, and meet the needs of users to a certain extent. The traditional network node load balancing and cache strategy restricted the function of the central network to a certain extent. Therefore, this article proposed a load balancing and cache management method for the central network, which was of great significance to accelerate the development of the central network. By analyzing the load scheduling law of network nodes, based on the existing load balancing and cache management methods, a cache node load balancing strategy suitable for the development needs of the central network was proposed, and a cache node information forwarding model for the central network environment was given. The results showed that the load balancing and cache management method proposed in this article can better meet the user needs of the central network and was superior to other traditional methods in the evaluation indexes such as load balancing degree, cache hit rate, and node response time. With the continuous development of communication technology and computer software and hardware, the research on load balancing and cache strategy of network nodes is also deepening. In the future, it is necessary to further optimize the load balancing algorithm and cache strategy to better adapt to the changes in central network structure and user needs.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Hebei Academy of Fine Arts.

References

[1] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos, “High performance, low complexity cooperative caching for wireless sensor networks,” Wireless Networks, vol. 17, no. 3, pp. 717–737, 2011.
[2] Y. Li, T. Jiang, K. Luo, and S. Mao, “Green heterogeneous cloud radio access networks: potential techniques, performance trade-offs, and challenges,” IEEE Communications Magazine, vol. 55, no. 11, pp. 33–39, 2018.
[3] F. Yin, A. Wang, D. Liu, and Z. Zhang, “Energy-aware joint user association and resource allocation for coded cache-enabled hetnets,” IEEE Access, vol. 7, pp. 94128–94142, 2019.
[4] S. Manfredi, F. Oliviero, and S. P. Romano, “A distributed control law for load balancing in content delivery networks,” IEEE/ACM Transactions on Networking, vol. 21, no. 1, pp. 55–68, 2013.
[5] Q.-V. Pham, T. LeAnh, N. H. Tran, B. J. Park, and C. S. Hong, “Decentralized computation offloading and resource allocation for mobile-edge computing: a matching game approach,” IEEE Access, vol. 6, pp. 75868–75885, 2018.
[6] Y. J. Sudha Rani and M. Seetha, “Enhanced POC tree-based algorithm for data item correlation and cache effective replacement in mobile ad hoc network,” International Journal of Pure and Applied Mathematics, vol. 118, no. 19, pp. 225–247, 2018.
[7] M. J. Siavoshani, F. Parvaresh, A. Pourmiri, and S. P. Shariatpanahi, “Coded load balancing in cache networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 2, pp. 347–358, 2020.
[8] M. J. Siavoshani, A. Pourmiri, and S. P. Shariatpanahi, “Storage, communication, and load balancing trade-off in
distributed cache networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 943–957, 2018.

[9] F. Liu, Z. Zhang, Z. Wang, and Y. Xing, “ECC: edge collaborative caching strategy for differentiated services load-balancing,” *Computers, Materials & Continua*, vol. 69, no. 2, pp. 2045–2060, 2021.

[10] S. Zhao, M. Hu, Z. Cai, Z. Zhang, T. Zhou, and F. Liu, “Enhancing Chinese character representation with lattice-aligned attention,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2021.

[11] M. Zghaibeh and N. U. Hassan, “D-SHAM: a constant degree-scalable homogeneous addressing mechanism for structured P2P networks,” *IEEE Access*, vol. 6, pp. 12483–12492, 2018.

[12] S. Lilly Sheeba and P. Yogesh, “Push-pull cache consistency mechanism for cooperative caching in mobile ad hoc environments,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, pp. 3459–3470, 2016.

[13] A. Larbi, L. Bouallouche-Medjkoune, and D. Aissani, “Improving cache effectiveness based on cooperative cache management in MANETs,” *Wireless Personal Communications*, vol. 98, no. 3, pp. 2497–2519, 2018.

[14] N. Karamchandani, U. Nieves, M. A. Maddah-Ali, and S. N. Diggavi, “Hierarchical coded caching,” *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[15] M. Xiang, Y. Jiang, Z. Xia, and C. Huang, “Consistent hashing with bounded loads and virtual nodes-based load balancing strategy for proxy cache cluster,” *Cluster Computing*, vol. 23, no. 4, pp. 3139–3155, 2020.

[16] J. Liao, K.-K. Wong, M. R. A. Khandaker, and Z. Zheng, “Optimizing cache placement for heterogeneous small cell networks,” *IEEE Communications Letters*, vol. 21, no. 1, pp. 120–123, 2017.

[17] F. Guo, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, “Joint optimization of caching and association in energy-harvesting-powered small-cell networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6469–6480, 2018.

[18] T. Ma, Y. Hao, W. Shen, Y. Tian, and M. Al-Rodhaan, “An improved web cache replacement algorithm based on weighting and cost,” *IEEE Access*, vol. 6, no. 16, pp. 27010–27017, 2018.

[19] A. M. Osman and N. I. Osman, “A comparison of cache replacement algorithms for video services,” *Int. J. Comput. Sci. Inform. Technol.*, vol. 10, no. 2, pp. 1–17, 2018.

[20] J. Wang, B. Berg, and D. S. Berger, “Maximizing page-level cache hit ratios in large web services,” *ACM SIGMETRICS - Performance Evaluation Review*, vol. 46, no. 2, pp. 91-92, 2018.

[21] S. Zhao, M. Hu, Z. Cai, and F. Liu, “Dynamic modeling cross-modal interactions in two-phase prediction for entity-relation extraction,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI-20 2020)*, pp. 1–10, Yokohama, France, 2021.