Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5970583, 16 pages
https://doi.org/10.1155/2020/5970583

Research Article

Network-Aware Data Placement Strategy in Storage Cluster System

Bilin Shao,[1] Dan Song,[1] Genqing Bian,[2] and Yu Zhao[1]

[1]*School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China*
[2]*School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China*

Correspondence should be addressed to Dan Song; 674355101@qq.com

Received 3 September 2019; Revised 13 January 2020; Accepted 10 March 2020; Published 21 April 2020

Academic Editor: Laurent Dewasme

The dramatic increase of storage devices in distributed storage cluster system and the inherent characteristics of distributed deployment mode make network resources become one of the bottlenecks of data storage process. By analyzing the functional characteristics of data flow transmission of the components of the storage system, the network topology structure of the storage system is constructed, and the evaluation index of node flow load is put forward based on the degree centrality and the betweenness centrality theory of the network to explore the network topology and real-time flow characteristics. According to the evaluation index of node flow load, a network-aware data layout scheme is proposed. By balancing the flow load of bottleneck link, congestion and transmission delay can be reduced to further shorten the total task execution time and improve the efficiency of data writing.

1. Introduction

In the large-scale data storage cluster system (hereinafter referred to as storage system), the shortage of network resources and the sharp increase in data flow are one of the main reasons for network congestion, slow data transmission, and service response delay in the storage system. In order to manage large-scale data access, it is undoubtedly an effective way to improve service response efficiency to accurately identify network characteristics and optimize data layout while monitoring the state of the whole network.

The network is the core support of the storage system and the bridge connecting all devices. In the data storage system, all system resources (storage devices, routers, switches, etc.) are connected to each other through network devices, which jointly constitute the network topology. Large-scale data storage systems are often built on a large number of cheap devices, and a quantity of data streams will be generated during the cooperation between device nodes. A storage device carries out data stream transmission with other devices through network link. Network bandwidth is a key index to measure the transmission capacity of the system. Because the network bandwidth is very limited, it is necessary to allocate the network bandwidth reasonably to improve the network transmission capacity.

Since the different network links and storage nodes carried out different data flows, the transmission characteristics of data flow in the network should be fully considered. Data flow is introduced as the weight to research network topology characteristics for accurately analyzing and identifying the storage system network characteristics. Due to the storage node location, degree, transmission capacity between data flow and link, residual capacity, transmission time interval, the transmission waiting queue (retention volume), the uplink input data flow, and the downstream output data flow of the network topology, all the factors are featured to evaluate the storage node load degree, which is easy to deploy the data and provides a good foundation to solve the problem of network congestion of flow engineering [1]. Therefore, in this paper, firstly the topology is built based on the characteristics of storage system to extract the statistical characteristics of the network data flow. Then, according to the network degree centrality and median centrality, the data node flow load evaluation

index is put forward. Finally, a data layout scheme is presented based on network conception, which treats the less network load node as the target position to reduce congestion waiting time, lower the delay of data transmission, and enhance the efficiency of the storage.

2. Related Work

2.1. Research Progress of Data Layout in Storage System. In recent years, research on data layout optimization mainly includes three aspects: computing power, storage power, and bandwidth optimization. Part of the research focuses on the optimization of physical resources (CPU, memory, and storage) by improving the performance of storage devices to enhance data access rate and data center efficiency. Another part of the research is based on the performance characteristics of the data center network topology to optimize the data layout and improve the efficiency of data transmission:

(1) The research on the data placement solution of storage system in the cloud environment mainly focuses on node evaluation, cost tradeoff, linear programming, content dependency, and other aspects. Node evaluation of target node placement is based on comprehensive consideration of the current disk space load conditions, size of the available storage, CPU ability, memory processing ability, disk I/O, communication bandwidth, traffic flow, response rate, and its historical access record and failure record. Node evaluation usually selects the appropriate node placed data file and its replicas according to the key attributes such as dynamic number of replicas and data popularity. [2–5]. The tradeoff between transport overhead, storage overhead, and user access performance [6–8] enables to gain maximum performance with minimal overhead. Based on this, the linear programming method is adopted to minimize the system cost [9–12] and reduce user response time and network load [13] by adding the consideration of network overhead. An overhead tradeoff layout solution can result in a good performance of load balance, but the transfer time is not optimal due to the damage of data dependency. Placing data blocks with high dependence in the same data center can reduce the transmission times across the data center as far as possible and further reduce the consumption of network bandwidth and the system delay responding to the task request in the system [14, 15]. Aiming at multilayer data center, a multilayer topology structure is constructed from the perspective of network planning, and a network bandwidth model is established to localize network traffic and reduce the communication overhead in the upper layer network switch, thus ultimately reducing the overall traffic across the data center and reducing the network consumption of the cloud data center [16, 17].

(2) In other application environments, such as distributed storage system [18–21], grid computing [22], online social network [23], peer-to-peer (P2P) network [24, 25], software-defined network (SDN) [26–28], content distribution network (CDN) [29, 30], and big data network storage [31, 32], the research on data layout management is of great value.

In the distributed storage system, the network aware repair framework based on the dependency between data, storage demand, and available bandwidth [18–20] can find the data repair scheme with the minimum bandwidth cost in the dynamic network and realize the load balance of storage and network traffic. Hedera [21] is a scalable dynamic flow scheduling system that schedules a multistage switching structure adaptively to utilize the resources of the aggregate network effectively.

In grid computing, the network-aware QoS workflow scheduling method [22] takes network characteristics and task dependence into account so that can reduce the completion time and workflow execution cost and improve the task success rate and resource throughput simultaneously. According to the characteristics of user interaction in social networks, the data placement method combining social graph division and data replication [23] divides users into a number of communities and further transforms the problem into a community-server secondary distribution problem according to the network topology of data centers.

In P2P networks, it is also an important direction to introduce the concept of replica population and apply knowledge of population ecology to solve data layout [24]. The distributed topology-aware unstructured P2P file caching architecture [25] can reduce the transmission traffic on the trunk network by caching hot data and reducing excessive caching of nonhot data.

In SDN, on the one hand, analysis of network real-time large data set to predict the future demand and realize the network traffic intelligent management decision [26] and, on the other hand, evaluate the status of network real-time by calculating the link bandwidth, delay, and packet loss rate to make route decisions dynamically, which can effectively realize load balance scheduling according to the estimation of traffic flow and link utilization ratio [27, 28].

In CDN, energy efficient delivery model (EEDM) [29] based on multicast tree can improve the scalability and uniform distribution of data storage to different degrees. The learning automata adaptive probabilistic search algorithm based on fuzzy support [30] makes use of the local topology information and current state of the cooperative nodes provided by the existing fog nodes and finds the point-to-point and point-to-fog minimum jumpers by running the distributed adaptive enhancement algorithm.

In big data network storage system, the continuous and uniform data striping layout method based on fragment label [33] and the discrete multireplica spatial data layout scheme based on graph coloring theory [34] can improve the scalability and uniform distribution of data storage.

(3) The optimization of virtual machine layout in the cloud environment has important inspiration for the research of data layout. The network-aware layout strategy adopted in the virtual machine layout in the cloud environment focuses on the traffic demand of the virtual machine and takes into account the quadratic and real-time variability of traffic, as well as the network topology and routing scheme [31]. By searching for the optimal bandwidth between average throughput and peak throughput, computing and network resources are allocated in a way that balances resource utilization efficiency and predictability of performance [32, 35], which solves the problem that the general network-aware VM layout scheme lack consideration of the optimal bandwidth allocation. The two-stage virtual machine placement algorithm of network awareness [36, 37] dynamically perceives the stability of the physical host according to the node centrality and the aggregation coefficient and appropriately aggregates virtual machines by the similarity, which improves the network communication capacity and reduces the network traffic between different data centers.

In summary, the data storage layout needs to take network resources into account significantly, and the network characteristics of the data center (topology, traffic characteristics, etc.) have an important impact on the performance of the data layout.

2.2. Application Scenario and Main Contributions of the Paper. Different data blocks of the same file in the storage system are often distributed and stored in different nodes on different racks. There are several storage nodes on each rack, and the nodes within the same rack are connected by Top-of-Rack (ToR) switch, and the nodes between different racks are connected by core network switch, as shown in Figure 1. Data transfer between intrarack nodes relies on ToR switch, and cross-rack data transfer depends on core network switch deployed in storage systems.

In storage systems, the link from the core network switch to the rack is the main network bottleneck [38, 39]. Each storage node in the storage system network can initiate data transmission through ToR to a storage node on the same rack or to a node on another rack with the core network.

At present, although cloud providers are deploying a large number of computing and storage devices to meet the growing demand for computing and storage resources, network resource demand is becoming one of the key factors for performance bottlenecks. In the storage system network, uneven flow distribution is easy to lead network congestion, and especially, flow load imbalance between bottleneck links is a major cause of network delay. Therefore, in network storage system, according to the characteristic of data storage network and complex network theory, the network flow distribution model is established and network flow concentration degree index and node centrality index of storage system are put forward to identify the characteristics of the network flow, implement the effective control, and balance network flow between multipath, which have very vital significance on reducing congestion and transmission delay.

In view of the current situation of insufficient bandwidth allocation research and optimization in data layout, considering the key role of network bandwidth in the storage system, this paper starts with the network topology structure and flow transmission characteristics and puts forward the evaluation index of node flow load and the data layout scheme of network awareness. Firstly, according to the characteristics of the storage system network, the data transmission between nodes is divided into cross-rack transmission and intrarack transmission, and the data center network topology is established. Secondly, by analyzing the real-time characteristics of the network topology in the storage system, the importance and load status of the nodes in the network topology are perceived from four indexes: node strength, node capacity centrality, data quantity transmitted by the nodes, and concentration index of data flow of node. Then, the network topology characteristics are constructed to establish the data layout. Finally, simulation experiments are carried out to verify the superiority of the new network awareness data layout strategy in the completion time of transmission tasks.

The contributions of this paper are as follows:

(1) The evaluation index of node load considered on network topology and real-time flow is proposed. The characteristic of cross-rack transmission and intrarack transmission is constructed. Four characteristic indexes are proposed, including node strength, node transmission turnover, node capacity centrality, and concentration index of data flow of node, and the comprehensive evaluation index of node network load based on these four characteristics.

(2) The network awareness data layout scheme is proposed. The task is written according to the remaining number of data block father file. The real-time characteristics of the network are sensed based on the storage system network topology structure and comprehensive evaluation index of node network load to select target and place racks. Considering the nodes network load and storage load, the node is placed in the target rack to finally complete the data layout optimization of network awareness.

3. Evaluation of Node Flow Load of Storage System

3.1. Storage System Network Topology Construction and Flow Statistical Feature Extraction. Network awareness is the real-time monitoring of all elements performance of the entire

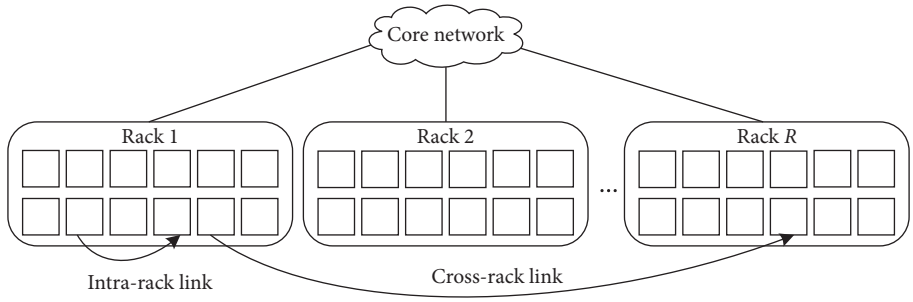

Figure 1: Hierarchy in data center.

network (network topology, network equipment, etc.) and the precaution and treatment of dynamic changes in network flow. In order to analyze the influence of network elements on the performance of data layout and identify the nodes with heavy load and the key nodes and intervals of data flow transmission, the attributes of key nodes and intervals should be considered from two aspects: network topology and the role of nodes and links in the process of data transmission.

(1) Network topology is to map various devices of the storage system to a node in the network. The network architecture in the storage system determines the role and influence of each node and link in the data transmission process and is an important factor to judge the real-time characteristics of the network. In general, the main network devices of storage systems include core network switches, ToR switches, and storage servers. According to the connection characteristics and transmission characteristics of these elements, this paper constructs a brief network topology diagram, as shown in Figure 2.

In order to facilitate modeling and simplify multilevel switch configuration, it is collectively referred to as core network configuration. In Figure 2, the node in the central position represents the core network, the dark gray node in the middle layer represents the overhead switch, and the light gray node on the edge represents the data storage server.

According to the established network topology, the node abstract method is adopted to construct the data center network topology diagram as G, and G is expressed as follows:

$$G = (V, E). \tag{1}$$

In formula (1), V represents the collection of all nodes in the network (routing nodes and storage nodes), and V is expressed as follows:

$$V = \left\{ v_i \,\middle|\, i = 1, 2, \ldots, N \right\}. \tag{2}$$

E represents the collection of connecting edges between switches or between switches and storage nodes. V is expressed as follows:

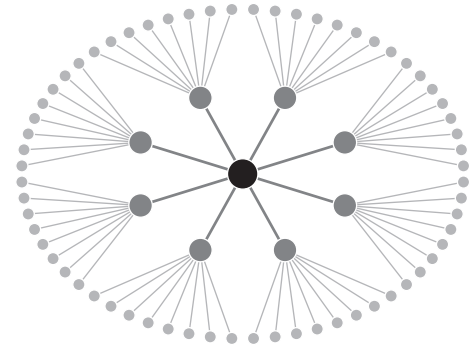

Figure 2: A typical network topology of storage system.

$$E = \left\{ e_{ij} \,\middle|\, i, j = 1, 2, \ldots, N, i \neq j \right\}. \tag{3}$$

A switch and a server correspond to node v_i in G, respectively, and the connection between the server and the switch serves is seen as edge e_{ij}.

(2) The role of nodes and links in the network during data transmission. Various network devices (switches, routers, etc.) in the network topology of storage systems play different roles in the data transmission process and have different importance. According to the importance and ability of network elements, the strength and importance of its role in data transmission are identified, and the data layout is carried out dynamically to ensure the strong service ability of core elements and improve the average utilization rate of common elements, which is crucial to improve the efficiency of the overall network.

According to the constructed network topology, the statistical characteristics of network flow are extracted. With each node v_i as a unit, all flow information passing through v_i is counted, including data flow information starting from v_i, that ending in v_i, and that passing through v_i, and current real-time transmission rate and maximum data transmission amount on each link. A tuple is defined to reserve the data flow information of each node (data amount initiated by the node, data amount received by the node, and data amount transferred by the node) and

node adjacent link information (link capacity and real-time used capacity). The data flow information of node v_i is represented by DataFlow$_i$:

$$\text{DataFlow}_i = \left([f_{si}, f_{ei}, f_{ti}], \left[\text{MC}_{ij}, \text{RTT}_{ij} \right] \right). \quad (4)$$

In formula (4), f_{si} represents the data amount starting from v_i, f_{ei} represents the data amount ending to v_i, f_{ti} represents the data amount transiting v_i, and v_j is the node connected to v_i; that is, for $v_j \in V$, $e_{ij} \in E$. MC$_{ij}$ and RTT$_{ij}$ are, respectively, link capacity and real-time used capacity of node adjacent link.

(3) Calculation of node distance. In the storage node distance calculation method of Hadoop distributed file system (HDFS), it is stipulated that the distance between the same rack node equals 2 and the distance between the cross-rack node equals 4. This paper follows this rule, and the distance between different types of nodes is calculated as follows:

$$d_{ij} = \begin{cases} 2, & v_i, v_j \text{ in same rack}, \\ 4, & v_i, v_j \text{ in different racks}. \end{cases} \quad (5)$$

In formula (5), v_i and v_j are different nodes in the storage system; that is, $i \neq j$ and d_{ij} represent the distance between two nodes.

3.2. Definition and Calculation of Storage System Network Characteristic Indexes. Node importance indicates the pivotability of a node in the network. The higher the node importance is, the stronger the pivotability is, and the heavier the flow load is. In addition, nonpivotability nodes are also loaded differently due to task preferences. Therefore, considering the influence of network topology and real-time flow on node load, this paper comprehensively evaluates the load degree of nodes from the two aspects of node importance and real-time flow.

Firstly, from the perspective of network topology, the degree of nodes intuitively reflects the importance of nodes in the network, and the number of node capacity centrality reflects the pivotability of nodes in the whole network data flow transmission process.

However, the importance of nodes in the whole network does not fully reflect the amount that data carries. Generally, the higher the importance of a node is, the more the data transmission tasks it carries, and the heavier the load is. However, on the one hand, in the actual transmission tasks, due to task preference, the actual amount of data carried by nodes of equal importance will vary. On the other hand, it is the timeliness of transmission tasks; that is, the amount of transmission tasks carried by each node in different time periods varies greatly. Therefore, the amount of data transmitted by the nodes directly reflects the amount of data carried by the nodes in the whole network data transmission, concentration index of data flow of node reflects the flow balance of the nodes in a certain period of time, and the network characteristic indexes are defined and explained according to the network topology structure constructed in the previous section.

3.2.1. Node Strength. The strength of the weighted network central node is defined as the sum of the weights of all the edges associated with the node. For the storage system network carrying data transmission flow, the strength of the node is the sum of the data flow of the corresponding zone cross-section. The calculation method is shown in equation (6). Node strength index mainly reflects the importance of nodes from the local network:

$$\text{CS}(i) = \sum_{j \in V_i} w_{ij}. \quad (6)$$

In formula (6), w_{ij} is the cross-sectional data flow of connection node v_i and v_j.

3.2.2. Node Capacity Centrality. The node capacity centrality is the ratio of the sum of all the cross-sectional data flow on the shortest path passing the node and the sum of all the cross-sectional data flow on all shortest paths in the network. The capability centrality reflects the node pivotability to the whole network flow.

In the storage system network, capacity centrality of node v_i not only counts the number of path passing through node v_i for all shortest paths in the whole network but also assigns different weights to each shortest path, namely, the sum of the cross-sectional data flow on the path, so as to more truly reflect the capacity of nodes to carry data flow. The calculation method of node capacity centrality is shown in the following equation:

$$\text{CC}(i) = \frac{\sum_{s,t \in V, i \neq s,t} \left[\left(\sum_{e \in R_{st}} F_e \right) \cdot \varphi_i(st) \right]}{\sum_{s,t \in V, i \neq s,t} \sum_{e \in R_{st}} F_e}. \quad (7)$$

In formula (7), R_{st} is the shortest path between s and t, e is an interval of R_{st}, and F_e is the sum of the data streams of the upstream and downstream sections of interval e. In this paper, R_{st} is calculated by the Dijkstra algorithm, as follows:

$$R_{st} = \text{Dijkstra}(s, t). \quad (8)$$

The calculation method of F_e is shown as follows:

$$F_e = \sum_{i,j \in e} \left(w_{ij} + w_{ji} \right). \quad (9)$$

$\varphi_i(st)$ is calculated by formula (10), which is based on the relationship between v_i and R_{st}:

$$\varphi_i(st) = \begin{cases} 1, & i \in R_{st}, \\ 0, & i \notin R_{st}. \end{cases} \quad (10)$$

3.2.3. Amount of Data Transmitted by Node. The amount of data transmitted by node v_i in the storage system network refers to that multiplied by all data flows through node v_i

with the corresponding transmission distance. The calculation method is shown in formula (11). The data amount index of node transmitted mainly considers the importance of node in topology from the two aspects of data flow size and data transmission distance:

$$\mathrm{CT}(i) = \sum_{i \in V} f_i \cdot d_i. \tag{11}$$

In formula (11), f_i is the data flow through node v_i and d_i is the transmission distance of the corresponding data. f_i mainly consists of three parts: f_{si} is the data amount with the starting point of node v_i, f_{ei} is the data amount with the end point of v_i, and f_{ti} is the data amount with v_i as the transition node; d_{si}, d_{ei}, and d_{ti} are the transmission distance corresponding to the transmission process, and then formula (11) can be further transformed into the following equation:

$$\mathrm{CT}(i) = \sum_{i \in V} \left(f_{si} \cdot d_{si} + f_{ei} \cdot d_{ei} + f_{ti} \cdot d_{ti}\right). \tag{12}$$

The relation between f_i and f_{si}, f_{ei}, and f_{ti} is shown as follows:

$$f_i = f_{si} + f_{ei} + f_{ti}. \tag{13}$$

3.2.4. Concentration Index of Data Flow of Node. HHI is a composite index to measure industrial concentration degree. This paper uses this concept for reference, puts forward CDF index (concentration index of data flow and the CDF index) of node v_i, and is defined as for a period of time squared as a percentage of the data flow that was passing on a node v_i. Calculation method is as shown in the following equation:

$$\mathrm{CDF}(i) = \left(\frac{f_i}{F}\right)^2. \tag{14}$$

In formula (14), f_i is all data flow passing through node v_i in a certain period of time which is calculated by formula (13) and F is the total amount of network transmission in the same period of time, which is calculated as follows:

$$F = \sum_{i \in V} f_i. \tag{15}$$

When all data are transmitted by one node, the data flow aggregation coefficient CDF(i) of that node is equal to 1. When all nodes are carrying the same amount of data transmission, CDF = 1/N2. The more data amount a node can carry, the greater the CDF.

3.2.5. Node Flow Load Comprehensive Evaluation Index (CEI). The previously defined node strength CS reflects the network node important degree under the different data flow states. The node capacity centrality (CC) reflects the data flow capacity that the node loaded. The node transmission data amount CT reflects the importance of the node in the entire network data transmission. Concentration index of data flow (CDF) of node reflects node flow balance status for a certain period of time. In order to facilitate the comparison, a comprehensive evaluation index (CEI) was defined and the above four indexes were integrated to collectively judge the importance degree and flow load status of the nodes. Since the dimension of each index is different, each index variable data are firstly standardized and converted into dimensionless values of CS′, CC′, CT′, and CDF′, and then they are given weights λ_1, λ_2, λ_3, and λ_4, respectively. The calculation method of CEI_i is shown in the following equation:

$$\mathrm{CEI}_i = \lambda_1 \mathrm{CS}' + \lambda_2 \mathrm{CC}' + \lambda_3 \mathrm{CT}' + \lambda_4 \mathrm{CDF}'. \tag{16}$$

Different networks focus on different needs; therefore, the appropriate weight value is chosen to meet different needs. For example, to fully evaluate the significance of a node in the entire network, then $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$. To evaluate the core position of node in the transmission of data flow in the whole network, the coefficient λ_3 of CT, such as $\lambda_3 => \lambda_1 = \lambda_2 = \lambda_4$, is increased to achieve the comprehensive ranking of all nodes in the whole network meeting the management requirements. In addition, there are three methods to determine the weight: subjective weighting method (such as expert survey method and hierarchy analysis process), objective weighting method (such as principal component analysis method, entropy method, and multiobject planning method), and combined weighting method ("multiplication" integration method and "addition" integration method).

3.3. Index Application and Result Analysis. For CEI proposed above, the topology structure containing 64 nodes is taken as an example for testing, and the corresponding topology structure is shown in Figure 3.

The data transmission task quantity was set as 500 files, and the data flow through each node was counted. According to the corresponding formula, the node strength, capacity centrality, data amount transmitted, and concentration index of data flow are calculated. Finally, the comprehensive evaluation index (CEI) is figured out, and the result graph is drawn.

For the topology structure mentioned above, different amounts of data transmission task (DF = 500) are produced. In four times, the flow load on each link is extracted, and at a certain moment, each node data amount is detected. The four indexes of each node are calculated as CS, CC, CT, and CDF. After normalization of data, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$, and CEI is figured out. According to the load index value of each node, the load of each link and the corresponding node is plotted, as shown in Figure 4. The darker the node color is, the larger the size is, indicating the heavier the load of the node is. Correspondingly, the larger the link width is, the heavier the load of the link at this moment is.

As shown in Figure 4, the load of each node and link varies at different times. The link load with dark color and large width is large, and the color and size of corresponding node is large; that is, the CEI value is large. The CEI value of the node in the central position is always large, indicating that the node plays a pivotal role in the network and carries a heavy load of data. The CEI value of the node at the edge is

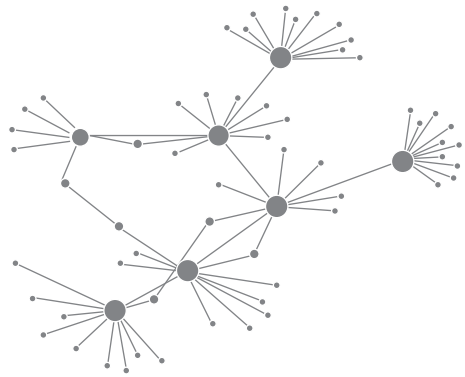

Figure 3: A network topology of storage system with 64 nodes.

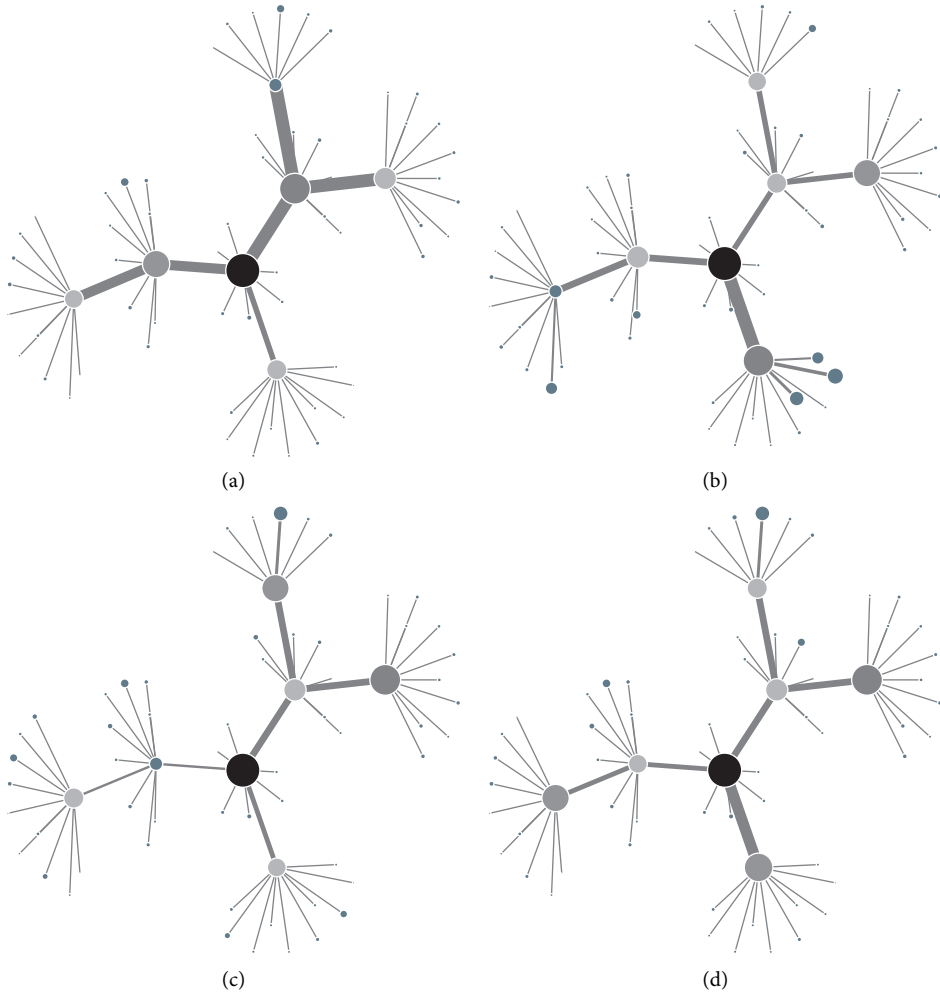

Figure 4: Link load and node loads at different times in the 64-node topology of storage system: (a) time $t1$; (b) time $t2$; (c) time $t3$; (d) time $t4$.

generally small since they are not responsible for forwarding data flow and other tasks, the importance of the node is low, and its value is mainly determined by the flow size of the associated link. Therefore, CEI proposed in this paper can comprehensively reflect the node importance and flow load characteristics in the storage system network topology.

4. Network Awareness Data Layout Scheme

4.1. Design Target. Generally, most of the cross-rack link load in a short period is more than two-thirds of the total load of links, which has increased the impact of a congestion link. If there is a data block of a file that needs to

communicate through congestion bottleneck link, the data block transmission progress will directly affect the entire file data transmission completion time; namely, the duration of the file transmission is completed by the slowest subfile data block transmission time.

During data writing, bottleneck links are almost always the hot spots. Considering the load condition of the network link in the storage system, the location selection and writing of different data blocks cut from a file are independent, and each data block is determined separately. Therefore, the main objectives of the data layout scheme design in this paper are as follows:

(1) Minimize the completion time of a single file. The optimal data block writing request sorting algorithm should consider the number of remaining blocks in the parent file of the data block. The data blocks with a small number of remaining blocks should be written first to speed up the completion of the transmission task of a single file.

(2) Minimize the unbalanced load of the bottleneck link. The optimal link selection algorithm should first eliminate the load imbalance on the bottleneck link and avoid too many transmission tasks concentrated on a small number of links, that is, the data to be written through the appropriate cross-rack link, so as to minimize the transmission delay.

(3) Minimize the unbalanced load of storage nodes. According to the flow load and space load of the storage node, the optimal layout algorithm should select the best target storage node for the arrived writing task so that the network load and space load balance effect of the storage node of the intrarack is optimal.

The mathematical description of the data layout problem discussed in this paper is as follows.

In the distributed cluster storage system, suppose there are a node set V that contains m data nodes $V = \{v_1, v_2, \ldots, v_m\}$ and a file set F to be written as $F = \{f_1, f_2, \ldots, f_k\}$. All k files will be stored in node set V, and data layout strategy is to assign these k files to m data nodes that achieve the optimal target function.

Three target functions are discussed in the network aware data placement strategy:

(1) Suppose the writing completion time of a single file as T_{single} and $T_{\text{single}} = t_{\text{finish}} - t_{\text{start}}$, where t_{start} is the start time of the file writing process and t_{finish} is the end time of writing to the file. It takes the least time for completing a single file writing task with the least value of T_{single}.

(2) Bottleneck link load balancing can be measured by network load changes in rack nodes. Standard deviation is appropriate for measuring the dispersion degree of data; it is consistent with the dimension of data, so the load balance of the rack node can be expressed by the standard deviation of load and used as the standard to measure the load balance of the system. The smaller the standard deviation of the load is, the better the load balancing ability is.

The load balancing law of bottleneck link LV is defined as

$$\text{LV} = \sqrt{\frac{\sum_{j=1}^{m} (\text{CEI}(j) - \overline{\text{CEI}})^2}{m-1}}. \tag{17}$$

In formula (17), $\overline{\text{CEI}}$ is the average of system load, $\overline{\text{CEI}} = (1/m) \times \sum_{j=1}^{m} \text{CEI}(j)$, and $\text{CEI}(j)$ is the traffic load of node v_j.

(3) The load balancing of the storage node is denoted as L. The storage load of the data node D_j can be calculated by the sizes of files that are stored in it, and$L(D_j)$ is calculated with the following equation:

$$L(D_j) = \sum_{i=1}^{n} S_k. \tag{18}$$

In formula (18), S_k is the size of all files on D_j.

Similarly, the standard deviation of the storage node load in each rack $L(R)$ is used to represent the rack load balance. The better performance of rack load balance is interrelated with the smaller $L(R)$. The calculation of $L(R)$ is shown as follows:

$$L(R) = \sqrt{\frac{\sum_{j=1}^{m} \left(L(D_j) - \overline{L}\right)^2}{m-1}}. \tag{19}$$

In formula (19), $\overline{L}$ is the average of system load, and $\overline{L} = (1/m) \times \sum_{j=1}^{m} L(D_j)$.

Therefore, the objective optimization problem of data layout can be represented by the mathematical model of the following equation:

$$\begin{cases} \min T_{\text{single}}, \\ \min \text{LV}, \\ \min L(R). \end{cases} \tag{20}$$

4.2. Network Awareness Data Layout Strategy. When the storage system is writing data, it first divides the data into several data blocks of the same size, and then the writing job of a file is divided into the writing task of several data blocks. To get the best file writing efficiency, it needs to optimize the completion time of each task. The main goal of data block writing in the storage system is to increase the writing rate of a single file by balancing the load on the bottleneck link to minimize the writing time of the data block. The optimal layout algorithm must allocate the best target location for the block writing request to let it pass through the appropriate bottleneck link.

In order to simplify the model, the following assumptions are made for the above analysis:

(1) The size of the data block to be written is fixed. Assuming all blocks are the same size, the impact of

the data block size difference on writing time is ignored.

(2) During the writing of a single data block, the link state is fixed. Assuming that the link utilization remains stable for a short period of time, it is easy to get the bottleneck link utilization very clearly during the entire data block writing process.

(3) The bottleneck link is easy to identify. In the storage system, the link between the rack and the core network is often the easiest and is most likely to become the bottleneck link. Therefore, this paper believes that the network bottleneck link is the link of in and out rack, that is, the dark link in Figure 2.

(4) Decision-making process of different data block layouts is independent. There is no impact between the writing decision processes of the last data block and the next data block, and they are independent.

On the one hand, the network awareness copy placement scheme needs to be sorted according to the arrival of data block requests; on the other hand, it needs to select appropriate links and target nodes, so the scheme contains the following three stages:

(1) Sorting of data block writing requests

The interval time between the two data block layouts is set as the decision time of the writing request sorting, denoted as s. The data block writing request arriving in the s decision time is sorted according to the number of remaining blocks in the parent file. To ensure the speed at which a single file transmission task can be completed, the smaller the number of remaining blocks is, the higher the ranking is. When s is equal to 0, it means that the layout scheme is an online decision-making process without the sorting process, which is processed directly according to the arrival order of data block writing requests.

The s value of the decision duration time determines whether there is the sorting process of data blocks to be written; that is, the data blocks to perform link selection and allocation will affect the layout decision of data block. The larger the s value is, the better the sorting result will be obtained by the algorithm, but at the same time, it will increase the writing time of the data block. Therefore, the value of s is a compromise process.

(2) Evaluation and sorting of rack loads

In Δt time interval, the current load data of all cross-rack links are obtained. Based on the evaluation index in Section 3.2, calculate the comprehensive evaluation index (CEI) of rack nodes and sorted rack nodes by CEI. The CEI is the basis for selecting the target rack. Rack with the least CEI having low traffic load will be the preferred target rack.

(3) Rack selection and storage node determination

The sorting result of load CEI of rack nodes calculated in the previous stage is read to take the rack with low CEI value as the target rack of data block writing request. In the target rack, according to the remaining space and flow load of the storage server node, two reachable server nodes with low load are selected as the target storage location.

The process of network awareness data layout is shown in Figure 5. Each dotted box in the figure represents the specific operation of each stage.

The process of network-aware data layout strategy is as follows:

Step 1: determine the order of block to be written. When the block write request arrives, the decision interval s is firstly determined. If $s > 0$, the ordering of written blocks is completed within the decision time s. In order to minimize the completion time of a single file, written block needs to sort in line with the number of remaining blocks in the parent file of the block. Blocks in the top with the least number of remaining blocks in the parent file, which may shorten the completion progress of writing a single file. If $s = 0$, block writing queen is sort by the "early come early service" principle to execute write operation.

Step 2: evaluate the rack node load. Cluster manager according to the received link transports information from each server node during Δt and updates the CEI value of rack node to maintenance load queue of rack node in time.

Step 3: select the target rack. The cluster manager allocates the target rack for the block to be written. The rack with the least network load is evaluated as the minimum CEI value, so the cluster manager chooses the rack with the least CEI value present as the target rack. During the Δt time interval, rack node with a lower CEI value is chosen for writing blocks and then the selected rack temporarily moved to the tail of the load queue until workload queue is updated at the next Δt time update.

Step 4: select the appropriate data node in target rack. The data nodes with less load are selected to place the data block in accordance with the load degree of the data nodes in the target rack. Network load LL and storage space load SL of data nodes in each rack are required. The load of each data nodes in the rack FF (n) is calculated to choose the data node with the minimum load as the target node for block placement.

4.3. Data Layout Algorithm of Network Awareness. According to the content and layout process of the three stages of the network awareness data layout strategy, the corresponding algorithms of the three stages are given below, as shown in Algorithm 1–3, respectively.

Algorithm 1 implements the sorting process of data block writing task. When s is equal to 0, the link selection operation is performed directly according to the arrival order of data block requests, or the sequence is sorted according to the number of remaining data blocks in the parent file of the data block, and the target rack and data

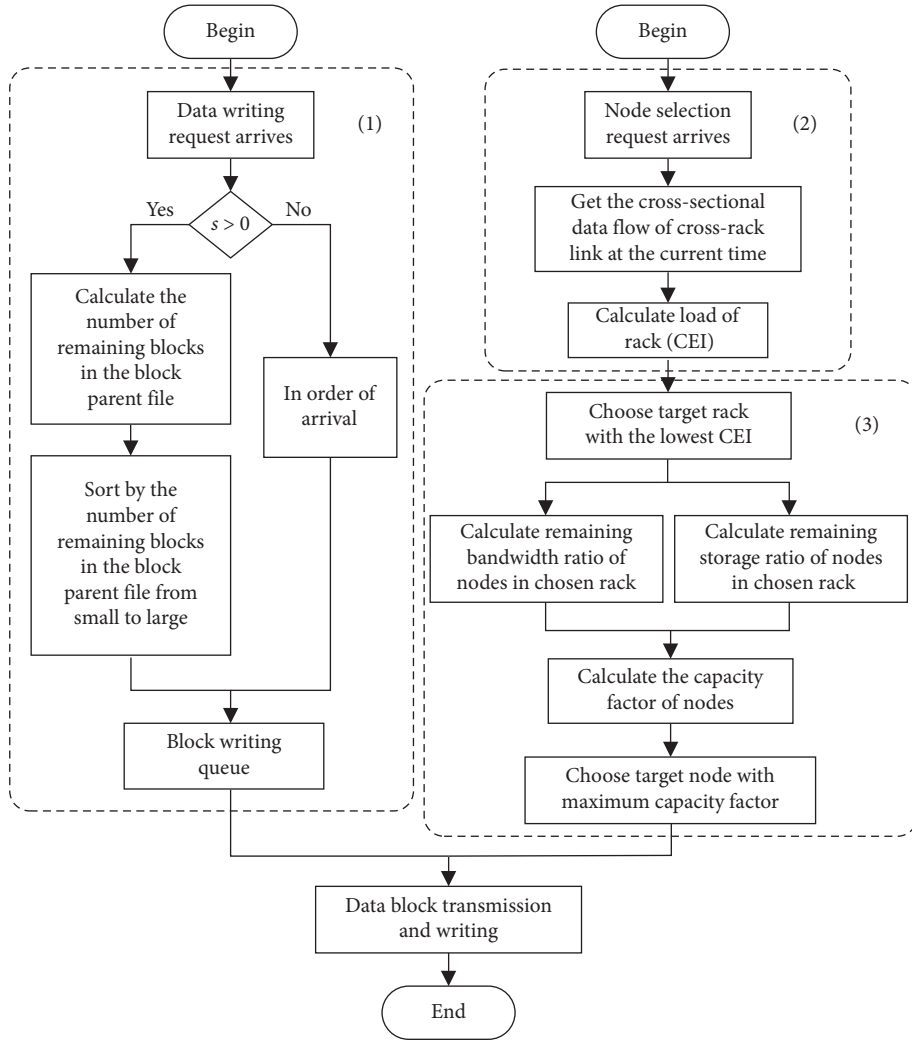

Figure 5: Network-aware data placement process.

Input: n nodes in rack Rr, link load, storage load
Output: data schedule queue Q
(1) Initialization: $D = \{d1, d2, \ldots, dm\}$
(2) if $s = 0$ then
(3) return LinkSelection(L)
(4) end if
(5) Q.addToQue(D) #add data block to queue
(6) Q.sort() #Order by policy
(7) for all data block d in Q do
(8) return LinkSelection(L)
(9) end for
(10) end

Algorithm 1: Request schedule algorithm.

node are selected firstly for the data blocks with a small number of remaining data blocks in the parent file.

Algorithm 2 firstly obtains the CEI value of the node according to the above calculation method and selects the rack with the smallest CEI. Link utilization assessment uses the information collected by the cluster manager (cluster topology, link load on the topology, and machine failure conditions) to make decisions.

The bottleneck link set Rr is composed of the links connecting the rack and the core network in the topology. CEI_r is used to express the current congestion degree of the link. The calculation method of the CEI is described in Section 3.2.

Input: L, link load
Output: link utilization; selected rack
(1) Initialization: require $NR = \{nr_1, nr_2, \ldots, nr_j\}$; $W = \{w_1, w_2, \ldots, w_j\}$; $F_e = \{F_1, F_2, \ldots, F_j\}$; $F = \{f_1, f_2, \ldots, f_j\}$, $d = \{d_1, d_2, \ldots, d_i\}$; F_{total}, λ_1, λ_2, λ_3, λ_4.
(2) for nr in NR do
(3) $\text{CS}_{nr} = \sum_{j \in V_i} w_{nrj}$
$\text{CC}_{nr} = (\sum_{s,t \in V, nr \neq s,t} [(\sum_{e \in R_{st}} F_e) \cdot \varphi_i(st)]) / (\sum_{s,t \in V, nr \neq s,t} \sum_{e \in R_{st}} F_e)$
$\text{CT}_{nr} = \sum_{nr \in V} f_{nr} \cdot d_{nr}$
$\text{CDF}_{nr} = (f_{nr}/F_{\text{total}})^2$
$\text{CS}'_{nr}, \text{CC}'_{nr}, \text{CT}'_{nr}, \text{CDF}'_{nr} = \text{uniformization}\ (\text{CS}_{nr}, \text{CC}_{nr}, \text{CT}_{nr}, \text{CDF}_{nr})$
$\text{CEI}_{nr} = \lambda_1 \text{CS}' + \lambda_2 \text{CC}' + \lambda_3 \text{CT}' + \lambda_4 \text{CDF}'$
(4) end for
(5) find the minimum CEI_{nr}
(6) return rack nr corresponding to CEI_{nr}
(7) end

Algorithm 2: Node load evaluation and selection algorithm.

Input: n nodes in rack Rr, link load, storage load
Output: the optimal node for placing one chunk
(1) Initialization: $Rr = \{N1, N2, \ldots, Nn\}$; $\text{FF}n = \{\text{FF}1, \text{FF}2, \ldots, \text{FF}n\}$
(2) for each node in Rr do
(3) $\text{SL}(n) =$ storage size of N/total storage capacity
(4) $\text{LL}(n) =$ link capacity from N to TOR/total link capacity from N to TOR
(5) $\text{FF}(n) = \text{SL}(n) + \text{LL}(n)$
(6) find the minimum $\text{FF}(n)$
(7) return data node with minimum $\text{FF}(n)$
(8) end

Algorithm 3: Node selection algorithm for link and storage load balancing.

The cluster manager receives link information from each server at regular intervals, including the load status of each link in the bottleneck link set. After receiving a single update, the utilization rate of each potential bottleneck link is calculated by the cluster manager. If the link information is missing, it is considered that the link is 100% utilized and has no available capacity; that is, the load factor is 1. At this time, transmission tasks are no longer assigned to the link.

Link updated time Δt decides the precision of the link information. Δt is smaller, the link updates at the higher frequency, and the result is closer to the current actual load. But if Δt is too small, it increased the load of cluster management server transmitting links. This paper uses the typical values of storage system $\Delta t = 1$ s [40].

Algorithm 3 calculates the load factor value of the node based on the storage load of each node in the selected rack and the link load from ToR to the node and selects the node with the minimum load factor value as the final placement location of the data block.

The network awareness data layout strategy has a certain delay. Once the writing request of a data block is accomplished, the evaluation value of the current utilization rate of all links involved in the transmission of the data block must be adjusted and updated in time to ensure the accuracy of subsequent layout decisions and avoid repeated decision results.

Expansibility description of network awareness data layout strategy: this scheme can be used in combination with some layout optimization strategies in the aspect of ensuring fault tolerance, partition fault tolerance, storage balance, and data reconstruction, so as to achieve better performance. For example, since the scheme in this paper focuses on the balance of flow load, if this scheme is combined with the layout scheme of storage balance, it can theoretically achieve better network balance performance while optimizing storage load balance.

4.4. Time Complexity of Network Awareness Data Layout Algorithm. For a given data node set V with size m, $V = \{v_1, v_2, \ldots, v_m\}$, file is set F to be written with size k, $F = \{f_1, f_2, \ldots, f_k\}$, and each file is divided into a number of data blocks to write. Suppose the number of individual racks is n, and the number of data nodes in each rack is m/n. Firstly, time complexity of sorting blocks to be written is the decision-making time s. The time complexity of calculating and finding the rack with the minimum load is $O(n)$, and the time complexity of calculating load of data nodes in rack and finding the data nodes with minimum load is $O(m/n)$. Maintenance of blocks writing queue and selection of rack and data node is executed concurrently, so take the worst time complexity of the two as the time complexity of the layout algorithm.

Therefore, the time complexity of the network-aware data placement algorithm is expressed as follows:

$$T = \text{Max}\left\{ks, O\frac{km}{n+kn}\right\}. \quad (21)$$

4.5. Functional Characteristics of Network Awareness Data Layout Algorithm. The core of the network-aware data layout strategy is to combine the load of the network link with the evaluation of the importance of the node to obtain more accurate node network load performance and then optimize the choice of target racks to balance network resources and reduce latency of data writing to reduce task completion time:

(1) The node load evaluation and selection algorithm can fully consider the importance of the node in the network topology and the real-time transmission of adjacent links to calculate the load of the node. Selecting a node with a smaller load as the target storage location can avoid assigning new transmission tasks to congested cross-rack links, thereby eliminating load imbalances of bottleneck link. In large-scale cluster storage systems, some links are prone to congestion in the network. The network-aware data layout strategy will select nodes with less link load to place data based on the node load evaluation results, avoiding selecting link with heavy transmission tasks to reduce task latency.

(2) The data block write request processing algorithm can minimize the completion time of a single file. According to the value of the decision duration s, different sorting strategies for writing data blocks are flexibly adopted. When $s > 0$, tasks can be sorted based on the number of remaining blocks in the parent file of the block. Files with a small number of remaining blocks are processed preferentially, which can shorten the writing completion time of a single file.

(3) The node selection algorithm for link load and storage load balancing can minimize the load imbalance of storage nodes. When selecting a storage node, the algorithm not only considers the load of storage space but also considers the network traffic load of the internal link of the rack. The target data node can be selected based on the network traffic load of the internal link of the rack, and the load of the data node storage space can obtain a better load balance of the storage nodes inside the rack.

(4) Maintaining the ordering of write task queues and node loads will increase task completion time. Firstly, sorting the write queue in time s will increase the task execution time. The larger the value of s, the better sorting result can be achieved, but at the same time, it will increase the data block write time. Therefore, take a suitable value s as an important process. Furthermore, updating link information takes Δt time and calculating the node value and selecting a node with a small CEI value also takes a little time, but the results have an important effect on balancing the link load and reducing task waiting time. Furthermore, the selection of storage nodes inside the rack consumes some time. The internal link load of the rack is lower than bottleneck links, so the time it takes to calculate and sort the load value of the storage node has negligible effect on the data writing time.

5. Experimental Evaluation

5.1. Experimental Setup. In the simulation experiment, the number of nodes was set as (1) 3000, including 150 racks, and each rack had 20 server nodes and (2) 300, including 15 racks, and each rack had 20 server nodes. The network topological structure of the storage system in the experimental test is shown in Figure 6, in which only 15 racks with a total of 300 nodes are drawn. Data transmission task number increases from 500, 1000, 1500, 2000, 2500, and 5000, respectively, and the experiment tests the data transmission completion time of the layout scheme in this paper at two states of normal link transmission congestion and link congestion. In this experiment, the size of data block is set to be the same.

The network was the only bottleneck set in the experiment. The cross-rack link is isomorphic with a maximum capacity of 10240 MB and so is the intra-rack link with a maximum transmission capacity of 256 MB. The transmission rate for the cross-rack link is 1024 Mb/s, and the transfer rate for the inner link of rack is 64 Mb/s. The initial load of the link is generated randomly, as well as the used space size of each storage node. The arrival rate of the data transmission task is 10 per second, the size of each data block is the same as fixed at 64 MB, and the transmission task is executed in the order.

In the test on the cluster storage system, HDFS cluster was built based on Hadoop 2.7.4 in the Linux environment, and three different cluster sizes were configured: (1) 1Master + 3DataNodes; (2) 1Master + 7DataNodes; and (3) 1Master + 11DataNodes. Firstly, the task completion time under different file writing tasks was tested by changing the number of file writing tasks, so as to analyze the performance when file writing load increased. Then, through changing the number of cluster nodes, the completion time of writing tasks for the same number of files under the three cluster sizes of 4 nodes, 8 nodes, and 12 nodes is tested, which is to analyze the impact on the performance of the layout strategy of cluster size.

5.2. Performance Effects of Network Status and Network Size. First of all, the experiment tested the layout of 15 racks with a total of 300 nodes and the data block transmission task with different numbers under noncongestion state of the link, counted the transmission task completion time under the network awareness data layout scheme, and measured the total transmission task completion time under the layout scheme without considering network load characteristics. The specific results are shown in Figure 7.

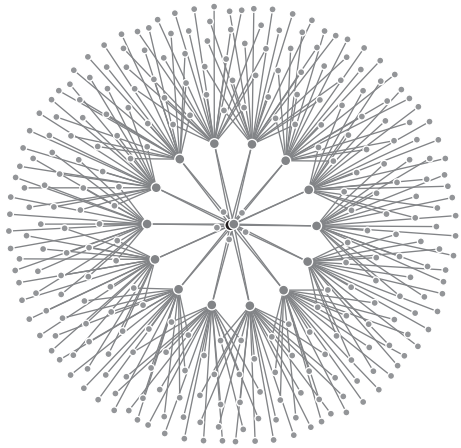

Figure 6: The network topology of the storage system under experimental test.

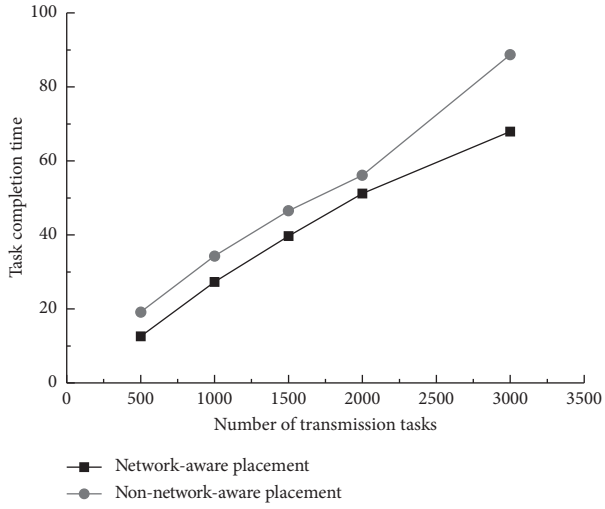

Figure 7: Task completion time of different schemes under noncongested network of 300 nodes.

Secondly, on the basis of the previous test, the congestion of the link is increased to test the total task completion time of the above two schemes in the case of different task transmission task quantities. The specific results are shown in Figure 8.

As shown in Figures 7 and 8, when the number of nodes is 300, the task completion time increases with the increase in congestion. In the condition of congestion, the data transmission task needs to wait for the link to be free before performing the transmission operation, so the waiting time is correspondingly increased, resulting in the increase in the total task completion time. The time for the network awareness layout scheme to complete the same number of transmission tasks is less than the execution time of the scheme without considering the network characteristics. On the one hand, the network awareness layout scheme avoids relatively more congested links and reduces the task waiting time. On the other hand, although the network awareness scheme costs sometime in the process of searching for high-quality nodes, it is found in the experiment that the time

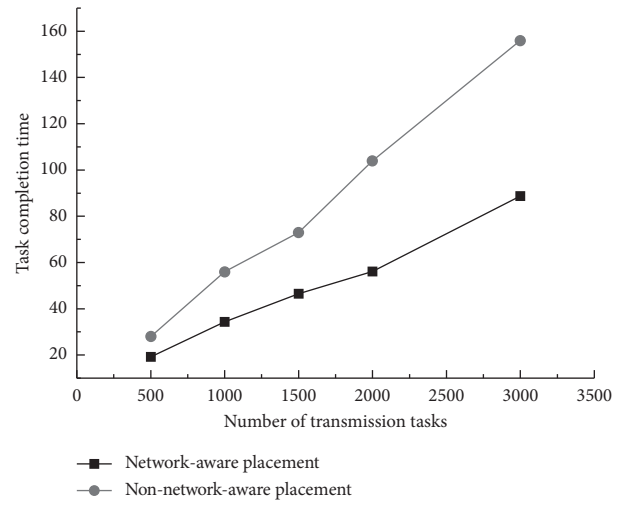

Figure 8: Task completion time of different schemes under congested condition of 300 nodes.

spent in searching for nodes in the topology with a small number of nodes is negligible.

Then, the topological network with a total of 3000 nodes of 150 racks was tested. Under the condition of relatively congested links, the total time for data transmission task of the above two layout schemes was measured, and the specific results are shown in Figure 9.

By comparing the results in Figures 8 and 9, the more the number of nodes increases, the more the total completion time of transmission tasks with the same number is. On the one hand, as the number of nodes increases, the time required for the node searching process adds, which results in an increase in the total time. On the other hand, as the number of nodes increases, for the scheme with no sensing, the possibility of repeatedly selecting the same node to store data is reduced, so the congestion is improved, and the change range of the total time to complete the task is smaller than that of the scheme with fewer nodes.

5.3. Performance Effects of the Number of Sort Policy. The simulation experiment tested the effect of different sorting strategies on the task completion time under 300 nodes that contain 15 racks. Under the noncongested network, change the value of s, respectively, as 0, 1, 2, and 5 to create four different blocks writing queue. The four different blocks writing queue include queue sorted by time of arrival, queue sorted by the remaining father file during $s = 1$ second interval, queue sorted by the remaining father file during $s = 2$ seconds interval, and queue sorted by the remaining father file during $s = 5$ seconds interval. Then, the task finish time of 500-block data transmission under the network-aware data layout method is tested and recorded. The specific results are as shown in Figure 10.

As shown in Figure 10, with the increase in file numbers (FNs), task completion time is in an upward trend. Firstly, by comparing the task completion time under $s = 0$ and $s > 0$, it is shown that the sorting algorithm did not significantly increase the time of data writing task at $s > 0$,

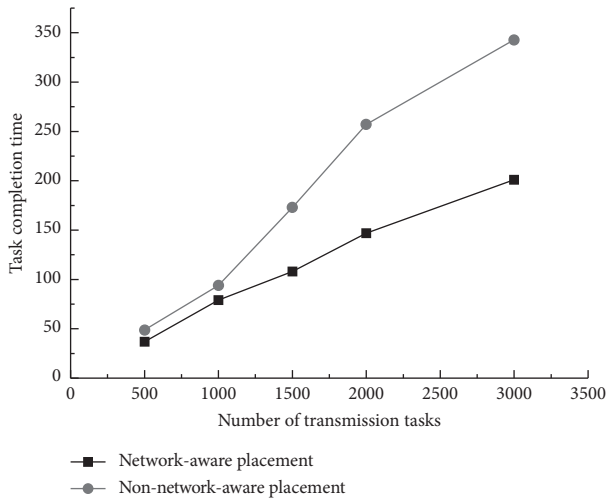

Figure 9: Task completion time of different schemes under congested condition of 3000 nodes.

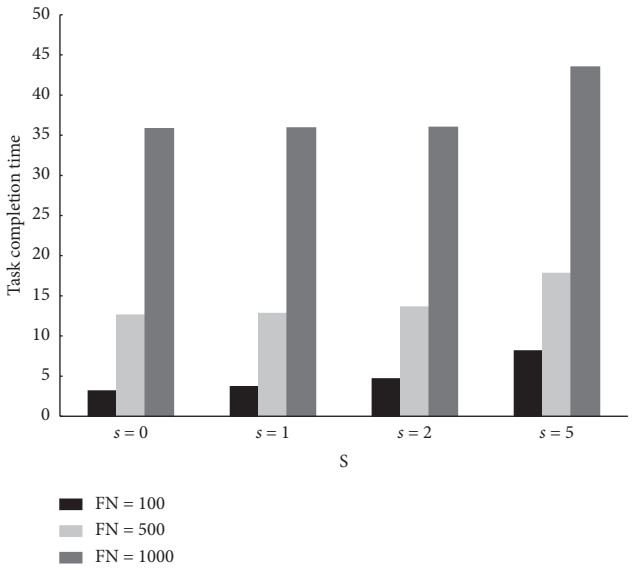

Figure 10: Task completion time of under different decision times s.

which indicates that the sorting decision had little impact on the completion time of data writing task. Then, we compared the completion time of written tasks under the decision times of $s = 1$, $s = 2$, and $s = 5$; it is shown that the task completion time when $s = 5$ was significantly higher than that $s = 1$ and $s = 2$. As stated in Section 4.2, the larger the value of s, the better the sorting results will be obtained by the algorithm, but the data block writing time will be increased at the same time. Therefore, the value of s is a compromise process. In this experiment, $s = 1$ and $s = 2$ are two suitable values.

5.4. Performance Effects of Cluster Size and Workload. The test results on cluster storage system of the network-aware data layout algorithm are shown in Figures 11 and 12. The performance of data layout algorithm under

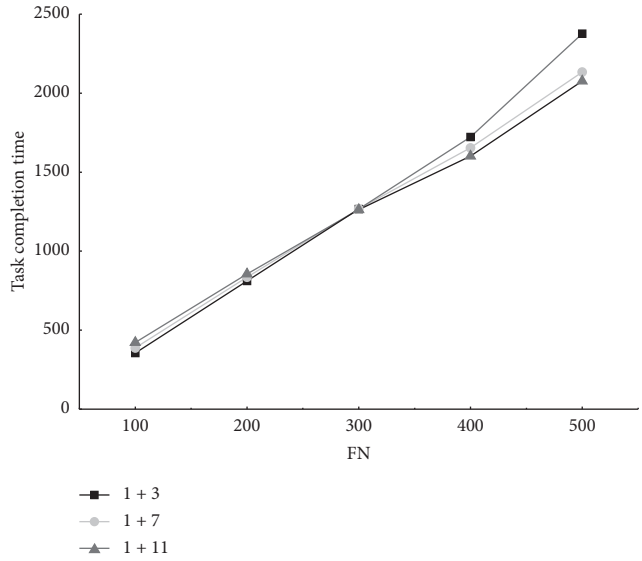

Figure 11: Task completion time under different cluster sizes.

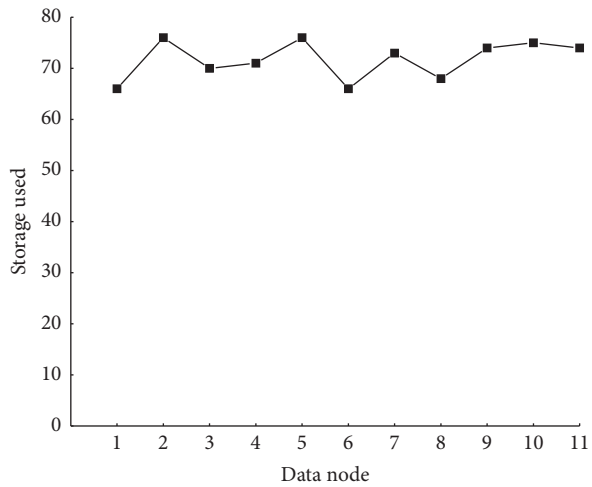

Figure 12: Storage load of each data nodes while FN = 100.

different scales was tested in HDFS. By increasing the number of cluster nodes to expand the cluster size, record the task completion time of the algorithm when FN = 100.

As shown in Figure 11, with the increase in the number of cluster nodes, the effect of the network awareness algorithm becomes better and better. As the number of files increases, the number of tasks to be transmitted increases, and the network load of the cluster storage system increases so that some link congestion is likely to occur. Network-aware data layout algorithm can avoid links with heavy transmission tasks and select nodes with less load to place data, thus reducing the task waiting time. However, as the number of tasks continues to increase, almost every link is saturated and the new writing task is added to waiting queue, and the performance of the network-aware data layout algorithm dropped because no matter which link is selected, blocks need to wait in this case.

In the test, the storage space load of 11 data nodes in the 1master + 11data nodes cluster was obtained under FN = 100 write task was completed, as shown in Figure 12.

As shown in Figure 12, the storage load of 11 nodes fluctuates between 65 and 80, which indicates that the algorithm has a good storage balancing effect. There are still some nodes with large load differences, such as nodes 2 and 6, because the load of storage space is not only considered in the selection of nodes but also the network traffic load of the internal link of the rack. The storage load of the cluster indicates that Algorithm 3 has a good load balancing effect in selecting the target data node based on the internal link network traffic load and the data storage space load.

6. Conclusion

Storage system network features will change significantly after carrying the data flow. Based on the complex network theory, this paper analyzes centrality index of storage nodes under the influence of storage system network data flow to identify node important degree, bearing capacity, and the equilibrium condition in the process of the storage system data transmission and further data layout performance optimization. Firstly, considering the local characteristics of data transmission, the path selection of data transmission, the distance of data transmission, and the carrying capacity of the nodes, four indexes of node strength, ability betweenness, data transmission amount, and concentration index of data flow are proposed for the identification of the node flow load in the data flow network. Then, according to the arrival time of the task and the data amount of the remaining blocks in the parent file of the data block, a flexible sorting method of the data block writing task is proposed. Finally, according to the result of node flow load identification, the target rack and storage node are selected according to the principle of least load, and a network awareness data layout scheme is proposed.

Experimental results show that the proposed data layout scheme of network awareness in this paper is better than that without considering the network characteristics of the layout plan in the aspect of transmission task completion time to improve the efficiency of data transmission task execution, reduce task execution time, effectively enhance the efficiency of data storage, and achieve the effect of network flow equilibrium. In the future research work, the network awareness data layout scheme based on future flow prediction will be further studied.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

D. S. designed the algorithms and wrote the paper. B. S., G. B., and Y. Z. made a careful revision of the article and proposed amendments.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61672416, the National Natural Science Foundation of China under Grant no. 61872284, and the Shaanxi Natural Science Foundation of China (2018JM6105).

References

[1] M. Shojafar, N. Cordeschi, D. Amendola et al., "Energy-saving adaptive computing and traffic engineering for real-time-service data centers," in *Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 1800–1806, IEEE, London, UK, June 2015.

[2] D. Jianguang, Z. Yuelong, and Y. Huaqiang, "Dynamic data replication management strategy in cloud computing environment," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 43, no. 10, pp. 53–57, 2015.

[3] Y. Lina, "Improvement of HDFS balanced placement strategy," *Computer Science*, vol. 44, no. S2, pp. 397–399+431, 2017.

[4] H. Dongmei, D. Yanling, H. Qi et al., "Marine monitoring data replica layout strategy based on multiple attribute optimization," *Computer Science*, vol. 45, no. 6, pp. 72–75, 2018.

[5] T. Yongcai, B. Yang, S. Lei et al., "Management mechanism of dynamic cloud data replica based on availability," *Journal of Chinese Computer Systems*, vol. 39, no. 3, pp. 490–495, 2018.

[6] W. A. Xiuguo, "Replica strategy considering cost and storage space in cloud environment," *Computer Engineering*, vol. 44, no. 3, pp. 19–26, 2018.

[7] L. Jun and H. Mengshu, "Replica placement strategy based on glowworm swarm optimization," *Application Research of Computers*, vol. 36, no. 2, pp. 584–587, 2019.

[8] Z. Bang, W. Xingwei, and H. Min, "Intelligent multiple data replica placement scheme for cloud storage," *Journal of Frontiers of Computer Science and Technology*, vol. 8, no. 10, pp. 1177–1186, 2014.

[9] M. Barshan, H. Moens, S. Latre, B. Volckaert, and F. D. Turck, "Algorithms for network-aware application component placement for cloud resource allocation," *Journal of Communications and Networks*, vol. 19, no. 5, pp. 493–508, 2017.

[10] J. Xiao, B. Wu, X. Jiang, A. Pattavina, H. Wen, and L. Zhang, "Scalable data center network architecture with distributed placement of optical switches and racks," *Journal of Optical Communications and Networking*, vol. 6, no. 3, pp. 270–281, 2014.

[11] W. Xiuguo, "Research on minimum cost data replica distribution based on dynamic planning in cloud storage system," *Computer Engineering*, vol. 43, no. 7, pp. 29–37, 2017.

[12] M. Alicherry and T. V. Lakshman, "Network aware resource allocation in distributed clouds," in *Proceedings of the IEEE INFOCOM (2012)*, pp. 963–971, IEEE, Orlando, FL, USA, March 2012.

[13] W. Xiuguo, "Minimum-cost based data replication strategy in cloud computing environment," *Computer Science*, vol. 41, no. 10, pp. 154–159+190, 2014.

[14] L. Xuejun, W. Yang, L. Xiao et al., "Datacenter-oriented data placement strategy of workflows in hybrid cloud," *Journal of Software*, vol. 27, no. 7, pp. 1861–1875, 2016.

[15] W. Yan and W. Jinkuan, "A dynamic replication placement mechanism in cloud storage," *Computer Engineering and Science*, vol. 39, no. 9, pp. 1581–1587, 2017.

[16] M. H. Ferdaus, M. Murshed, R. N. Calheiros, and R. Buyya, "An algorithm for network and data-aware placement of multi-tier applications in cloud data centers," *Journal of Network and Computer Applications*, vol. 98, pp. 65–83, 2017.

[17] L. Qingying, X. Lin, and L. Xicong, "Energy efficient cloud data replica layout algorithm considering network bandwidth," *Science Technology and Engineering*, vol. 19, no. 5, pp. 172–178, 2019.

[18] A. Uta, O. Danner, C. van der Weegen et al., "MemEFS: a network-aware elastic in-memory runtime distributed file system," *Future Generation Computer Systems*, vol. 82, pp. 631–646, 2018.

[19] M. Sipos, J. Gahm, N. Venkat, and D. Oran, "Network-aware feasible repairs for erasure-coded storage," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1404–1417, 2018.

[20] A. Epstein, E. K. Kolodner, and D. Sotnikov, "Network aware reliability analysis for distributed storage systems," in *Proceedings of the 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, pp. 249–258, IEEE, Budapest, Hungary, September 2016.

[21] M. Al-Fares, S. Radhakrishnan, B. Raghavan et al., "Hedera: dynamic flow scheduling for data center networks," *NSDI*, vol. 10, no. 8, pp. 89–92, 2010.

[22] S. John and M. Mohamed, "A network performance aware QoS based workflow scheduling for grid services," *The International Arab Journal of Information Technology*, vol. 5, no. 15, pp. 894–903, 2018.

[23] Z. Jingya, F. Jianxi, and W. Jin, "Data placement approach for scalable online social networks (in Chinese)," *SCIENTIA SINICA Informationis*, vol. 48, no. 3, pp. 329–348, 2018.

[24] X. Meng, Y. Wang, and Y. Gong, "Perspective of space and time based replica population organizing strategy in unstructured peer-to-peer networks," *Journal of Network and Computer Applications*, vol. 49, pp. 1–14, 2015.

[25] G. Gao, R. Li, H. He, and Z. Xu, "Distributed caching in unstructured peer-to-peer file sharing networks," *Computers & Electrical Engineering*, vol. 40, no. 2, pp. 688–703, 2014.

[26] S. K. Bhatti, M. I. U. Lali, B. Shahzad, F. Javid, F. U. Mangla, and M. Ramzan, "Leveraging the big data produced by the network to take intelligent decisions on flow management," *IEEE Access*, vol. 6, pp. 12197–12205, 2018.

[27] L. Qi, W. Lu, Y. Xiao et al., "Path selection algorithm based on open daylight network awareness and user requirements," *Journal of Chinese Computer Systems*, vol. 39, no. 8, pp. 1737–1743, 2018.

[28] R. Wang, S. Mangiante, A. Davy et al., "QoS-aware multipathing in datacenters using effective bandwidth estimation and SDN," in *Proceedings of the 2016 12th International Conference on Network and Service Management (CNSM)*, pp. 342–347, IEEE, Montreal, Canada, November 2016.

[29] L. Yujie, L. Dianjie, and Z. Guijuan, "Cloud content delivery network based on energy optimization," *Journal of Chinese Computer Systems*, vol. 39, no. 10, pp. 2216–2221, 2018.

[30] M. Shojafar, Z. Pooranian, and P. G. V. Baccarelli, "FLAPS: bandwidth and delay-efficient distributed data searching in fog-supported P2P content delivery networks," *The Journal of Supercomputing*, vol. 73, no. 12, pp. 5239–5260, 2017.

[31] O. Biran, A. Corradi, M. Fanelli et al., "A stable network-aware vm placement for cloud systems," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 498–506, IEEE, Ottawa, Canada, May 2012.

[32] R. Wang, J. A. Wickboldt, R. P. Esteves, L. Shi, B. Jennings, and L. Z. Granville, "Using empirical estimates of effective bandwidth in network-aware placement of virtual machines in datacenters," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 267–280, 2016.

[33] F. Dongyu, Z. Ligu, X. Zida et al., "Approach for optimizing data placement on mongo DB cluster," *Computer Engineering and Applications*, vol. 53, no. 17, pp. 77–84, 2017.

[34] L. Shengbin, T. Xiaoming, L. Zhiqing et al., "Discrete layout strategy for multiple replica of spatial data based on parallel computing," *Journal of Jilin University (Science Edition)*, vol. 54, no. 3, pp. 524–528, 2016.

[35] R. Wang, R. Esteves, L. Shi et al., "Network-aware placement of virtual machine ensembles using effective bandwidth estimation," in *Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 100–108, IEEE, Rio de Janeiro, Brazil, November 2014.

[36] W. Xiaojie, X. Mingwei, and W. Sixiu, "Two-phase virtual machine placement algorithm based on network awareness," *Computer Engineering*, vol. 43, no. 8, pp. 32–37, 2017.

[37] C. Lei, Z. Jing, and C. Lijun, "A network-aware two-phase virtual machine allocation algorithm," *Journal of Hunan University (Natural Sciences)*, vol. 43, no. 4, pp. 120–132, 2016.

[38] F. Ahmad, S. T. Chakradhar, A. Raghunathan et al., "ShuffleWatcher: shuffle-aware scheduling in multitenant MapReduce clusters," in *Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pp. 1–13, Philadelphia, PA, USA, June 2014.

[39] J. Li, S. Yang, X. Wang et al., "Tree-structured data regeneration in distributed storage systems with regenerating codes," in *Proceedings IEEE INFOCOM 2010*, pp. 1–9, IEEE, San Diego, CA, USA, March 2010.

[40] M. Chowdhury, S. Kandula, and I. Stoica, "Leveraging endpoint flexibility in data-intensive clusters," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 231–242, 2013.