Research Article

Haijie Shen*, Yangyuan Li, Xinzhi Tian, Xiaofan Chen, Caihong Li, Qian Bian, Zhenduo Wang, and Weihua Wang

Mass data processing and multidimensional database management based on deep learning

https://doi.org/10.1515/comp-2022-0251
received April 27, 2022; accepted July 24, 2022

Abstract: With the rapid development of the Internet of Things, the requirements for massive data processing technology are getting higher and higher. Traditional computer data processing capabilities can no longer deliver fast, simple, and efficient data analysis and processing for today’s massive data processing due to the real-time, massive, polymorphic, and heterogeneous characteristics of Internet of Things data. Mass heterogeneous data of different types of subsystems in the Internet of Things need to be processed and stored uniformly, so the mass data processing method is required to be able to integrate multiple different networks, multiple data sources, and heterogeneous mass data and be able to perform processing on these data. Therefore, this article proposes massive data processing and multidimensional database management based on deep learning to meet the needs of contemporary society for massive data processing. This article has deeply studied the basic technical methods of massive data processing, including MapReduce technology, parallel data technology, database technology based on distributed memory databases, and distributed real-time database technology based on cloud computing technology, and constructed a massive data fusion algorithm based on deep learning. The model and the multidimensional online analytical processing model of the multidimensional database based on deep learning analyze the performance, scalability, load balancing, data query, and other aspects of the multidimensional database based on deep learning. It is concluded that the accuracy of multidimensional database query data is as high as 100%, and the accuracy of the average data query time is only 0.0053 s, which is much lower than the general database query time.

Keywords: deep learning, data processing, MOLAP model, database management

1 Introduction

Driven by social development, a large amount of data has been established in the fields of agriculture, industry, transportation, medical care, and environmental protection. In the context of continuously deepening business and information, data from various industries will penetrate into various situations, that is, the company’s daily business applications. A large amount of data processing and analysis technology helps companies quickly and effectively understand changes in market conditions, make rapid decisions, and seize opportunities for development. At the same time, this application puts higher requirements on database technology.

A multidimensional database is a data warehouse that can perform data integration and calculation, as well as data retrieval and other functions. One of the
The innovations studied in this article are as follows: (1) The combination of quantitative analysis and qualitative analysis is fully reflected in the analysis process of the test in Section 4; (2) the combination of empirical analysis and theoretical analysis. This innovative method runs through this article.

2 Methods of massive data processing and multidimensional database management based on deep learning

2.1 MapReduce technology

The MapReduce data processing technology was proposed by Google, and it was originally designed to solve the problem of a large amount of data in its own search engine [4]. MapReduce takes traditional querying, decomposition, and data analysis and distributes the processing tasks to different processing nodes, thus providing greater parallel processing capabilities. MapReduce technology is divided into two steps. The first is the Reduce concept, where many things in our daily lives need to be expressed and output by the Reduce program; the second is the Map concept, which is mainly used for data manipulation and processing. The completed input and output process of MapReduce data processing technology is to perform Map or Reduce operations. Nowadays, machine types and programs are very different, and incompatibility problems are prone to occur. Therefore, under normal circumstances, it is necessary to partition and then block operations through Map first before proceeding to the next step. In general, reduce processing adheres to the user-specified area and task operations [5,6]. MapReduce is suitable for data analysis, log analysis, business intelligence analysis, customer marketing, large-scale indexing, and other operations and has very obvious results.

MapReduce is a computer model, framework, and platform for parallel processing of big data. It includes the following three concepts:

(1) It is based on a complex, high-performance parallel computing platform. This is a parallel market consisting of hundreds of computers, including hundreds of nodes [7]. Each node has its own memory, its own storage space, and a problem with one node does not affect the others.
(2) It is a parallel computing and functional software framework. The computer system can batch process data information by itself, and upload the processed data information to the cloud to form a complete database. The lowest layer allocates data to store more complex details. This will greatly reduce the tolerance of software and data processing [8].

(3) It is a parallel programming model and method. Use the design philosophy of the functional programming language Lisp, provide a simple parallel programming method, use Map and Reduce2 function scheduling to achieve basic parallel computing, and provide remote function and parallel interface programming to simply complete large-scale data planning and calculation processing [9].

2.2 Parallel database technology

Parallel database technology is the product of the fusion of parallel computing and database technology. The parallel database system is a new generation of high-performance database systems built on the basis of massively parallel processing and the cluster parallel computing environment. In order to improve the efficiency of data processing, people continue to realize that parallel processing of space and time can greatly improve efficiency. Work parallel programs and data parallel programs together form a parallel computer, and their roles are completely different. Regarding the management and adjustment of things, parallel processing makes it more complicated. On the contrary, the operation of parallel data has become more complicated. In order to facilitate processing, a large amount of work is divided into multiple sub-units. Performance and response time are performance indicators that measure the advantages and disadvantages of databases [10].

In order to design parallel databases, researchers need to improve the performance of both. The goal of the parallel database system is high performance and high availability, which improves the performance and availability of the whole database system by executing database tasks in parallel across multiple processing nodes. The architecture is the first design of public memory. Due to the structure of the shared disk and the design of the shared data system, not all processors can share a memory, so the communication efficiency is very high, and the memory access speed in access and data processing will also become faster. Usually, we choose this system architecture.

The goal of the parallel database algorithm is to improve the efficiency of the database, and the index is a method to improve the efficiency of the database [11]. In addition, data tend to become a general index creation program in a stable standard format, which greatly improves the efficiency of the database [12,13]. Figure 1 shows a flowchart of massive data processing.

The process of data processing is broadly divided into three stages: data preparation, processing, and output. In the data preparation stage, the main goal is to enter the data into the corresponding database. The data after entering are processed by the computer, for which the user prepares the program and inputs it into the computer, which then processes the data according to the program’s instructions and requirements and finally returns the results.

2.3 Distributed memory database technology

The distributed database is a product that combines past database technology and network technology. In this database, each shard is linked to one or more nodes, so it is less prone to a single point of failure. Distributed databases are scattered across the nodes of the computer network in the physical space, although they can logically belong to the same system for data collection. The decentralized memory database technology has local spatial autonomy, global reasonable sharing and use, data-rich security functions, data independence, and system transparency [14]. Distributed database management systems need to support global centralized control and distributed global control methods. This includes local site database management systems, global management systems, databases, global data dictionaries, and communication management. To be responsible for the creation and management of the local database: realize the autonomy of the website; execute local applications and other functions; provide distributed transparency; adjust the implementation of global events; adjust various local database management systems; and ensure the global consistency of the database, information synchronization, and other functions such as database technology, artificial intelligence technology, network communication technology, parallel computing technology, etc.

In this system, the following requirements must be met: (1) The database in the memory of each network
node maintains its autonomy. (2) The memory database needs to be compiled, read, and recorded in a separate way to solve the vertical and horizontal segmentation strategy of large-scale data storage. (3) Various data division methods based on the overall vertical division function of the horizontal division. Different applications and data need to be handled in different ways. (4) A node is a collection of logical computing resources that provide a unit of service. The in-memory databases of each node are adjusted mutually, and all the databases in the memory can be used as servers for other nodes. (5) Maintain the transparency of data dispersion, meet the characteristics of data dispersion and adjustment between databases, improve the balance between memory databases, and combine to solve the real-time processing requirements of a large amount of Internet of Things data. (6) To complete the level 2 asynchronous writing of the tolerance to the memory database and the change- ability of the memory database, it must be copied to the disk database and copied through the database.
2.4 Distributed real-time database technology based on cloud computing technology

In order to achieve real-time data and things in real-time systems, it is necessary to speed up the response and processing speed of real-time data as much as possible, but due to the weak real-time and unpredictability of the execution time of traditional database design I/O operations, buffer management, page violations, etc., real-time databases came into being, using the software group of cloud computing centers scattered around the world, the close combination of cloud-based shared real-time database, real-time database technology, cloud computing technology, data processing compression, data recovery, data storage imaging technology, competitive processing, and control system. Ensuring the scale and scalability of the database, the reliability of the real-time high-performance database management system and the sustainable dispersion database system include many functions such as digital delivery network technology, transaction planning, error monitoring and recovery, and cargo balance [15,16]. Based on real-time decentralization and virtualization, it performs large-scale data storage and synchronous transaction processing and other functions and handles storage encryption, distributed backup creation, and dynamic system expansion.

In the distributed real-time database architecture, the data collection nodes and service elements of the database server are connected to the platform through the media software interface of the distributed service platform (i.e., interactive communication with other service elements) [17]. Each element connects and calls other functional elements through maintenance methods to realize the freedom and effect of data interaction. In addition, data can also be sent and obtained through the interface of the distributed communication service platform through the communication link with other nodes that can access the service. The distributed communication service platform uses an internal temporary memory queue and an asymmetric call mechanism so that the node does not need to care about the status of the receiving node when sending data, and the node performs data acquisition by calling messages when receiving data. At present, distributed technology has become the core technology of modern computer information systems and application system development, which has become the core of the composition of computer information systems and application systems, and is one of the supporting technologies of the future information highway.

Many data storage and acquisition service components required by data collectors and servers are connected to the platform through cloud services to form integrated data storage and data recovery services and provide services to the outside to achieve the breakthrough of Alta data processing servers so far a unique island mode [18]. The distributed storage layer provides distributed storage capability for real-time data. The storage layer functions include the definition of data slicing, distributed data storage by means of clustering, and redundant mutual backup of data migration between nodes. A distributed server was formed, as well as the same kind of distributed data storage, data recovery, and other system functions. The data collector or data server needs to send the real-time collected data to the integrated storage service unit through the service platform. Store data in real time. The customer connects to the communication service platform through the platform interface or Web server, sends a request to the integrated data retrieval service, and executes the data request. For server nodes that send data to other nodes through the distributed communication service platform, the success of data transmission can be regarded as the success of data writing. After the node receives the data, it calls the interface to complete the data collection [19].

3 Experiments on massive data processing and multidimensional database management based on deep learning

3.1 Mass data fusion algorithm model based on deep learning

Typical models of deep learning include convolutional neural network models, deep trust network models, and stack auto-encoding network models, the most typical of which is the convolutional neural network model [20].

Before the pre-training mode of collective neural networks appeared, the training of deep neural networks was usually very difficult. A specific example is collective neural networks. The robust neural network is triggered by the construction of the optical system. This is the calculation of Fukushima’s first neural network model. Based on the local connection between neurons and multilevel body image conversion, neurons with the same
parameters are applied to the neural network at different positions in the upper layer, thereby changing the structure of the neural network. After that, based on this idea, LeCun et al. designed and trained a collective neural network with fault gradients and achieved high performance in certain standard recognition tasks. So far, the standard recognition system based on a comprehensive neural network, especially in handwritten character recognition tasks, is one of the best implementation systems [21,22].

The structure of the cable news network includes three convolutional layers: a concentration layer and two fully connected layers [23]. Before using convolutional neural networks model (CNNM) to export the data security function model, the model training has to be completed. The traditional training method is mainly the backpropagation algorithm. Due to the overall level and level, the CNNM model needs to be modified and focused. The CNNM training loss function is

\[
J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y^{(i)} \ln h_\theta(x^{(i)}) + (1 - y^{(i)}) \ln(1 - h_\theta(x^{(i)})) \right].
\]

The training goal is given by the following formula:

\[
\theta_t = \theta_{t-1} - \alpha \frac{\partial}{\partial \theta} J(\theta).
\]

In order to find the partial derivative for the convolutional layer, we have

\[
\delta^l_j = \beta^{l+1}_j \{ f'(u^l_j \text{up}) \delta^{l+1}_j \},
\]

where \( \delta^l_j \) represents the sensitivity of the \( j \)th feature map of the \( l \)th layer; \( \beta^{l+1}_j \) represents the parameter of the \( j \)th feature map of the \( l+1 \)th layer. Substituting \( \delta^l_j \) into the following two formulas, we obtain the convolution kernel weight \( w \) and the derivative of the bias \( b \) as follows:

\[
\frac{\partial J}{\partial w_{ij}} = \sum_{u,v} (\delta^l_j)^*w(p^{l+1}_{uv}),
\]

\[
\frac{\partial J}{\partial b_j} = \sum_{u,v} (\delta^l_j)^*b_{uv}.
\]

In the formula, it is the result of a convolution operation between the 1–lth layer feature map and the lth layer convolution kernel. At this point, equations (4) and (5) can be substituted into equation (2) to complete a parameter update of the convolutional layer.

For the pooling layer,

\[
z_j^l = f(\beta^l_j \text{down}(z_j^{l-1}) + b^l_j),
\]

\[
\delta^l_j = \sum_{j-1}^{m} \beta^{l+1}_j * k_8.
\]

In the formula (6), \( z_j^l \) represents the jth feature map of the Lth layer; down means to perform a pooling operation. The derivative of the weight and bias of the convolution kernel are obtained by the above two formulas, as shown in the following two formulas, and then the result is substituted into formula (2) to complete a parameter update of the pooling layer.

\[
\frac{\partial J}{\partial w_{ij}} = z_j^l * \delta^{l+1}_j,
\]

\[
\frac{\partial J}{\partial b_j} = \sum_{u,v} (\delta^l_j)^*b_{uv}.
\]

### 3.2 Multidimensional OLAP (MOLAP) model based on deep learning

The MOLAP function saves the processed source data and the results of problem analysis in a multidimensional database. Data are stored in multidimensional data before data storage. Can quickly answer the user’s questions and analysis requirements, and answer the questions with better answers to reply to the results. MOLAP can effectively handle the calculation of complex sizes and provide simulation analysis. However, in order to obtain data in real time, MOLAP needs to continuously update the data set, which is very costly. For data storage, MOLAP is limited by the volume of the hypercube defined by the multidimensional projection [24,25].

MOLAP belongs to the same memory data storage format as relational on-line analysis processing and hybrid on-line analysis processing. It mainly includes the following aspects: dimension (the catalog of data classification in the cube); hierarchy (the data in the dimension is from large to small, from macro to specific).

MOLAP cubes are used to store the results of calculations or problems. This includes dimension tables, including dimension tables, unary functions, and other members. In other words, it can provide users with this kind of information. In a specific dimensional environment, relevant information that is helpful for analysis and decision-making can be obtained from the measured value selected by the user. Multidimensional data sets include time dimension, location dimension, product dimension, and specific dimension member values and can return measurement values under a combination of specific sizes. For example, an element of the cube (1 January 2016, Guangzhou, coffee, ¥1,800) means that the sales of coffee in the Guangzhou area on 1
January 2016 was 1,800 RMB. Of course, if the aggregation algorithm is different, the cube data obtained are different, the query conditions are different, and the granularity level of the cube data obtained is also different. Relatively speaking, January 1, 2016, Guangzhou, Coffee, ¥1,800 represents fine-grained cube members, and January 1, 2016, South China, Coffee, ¥32,500 represents coarse-grained cube members. Based on the existing basic factual data, the user may have further query needs. For example, he not only wants to know the sales volume of a certain product in a certain area but may also want to know the product in a certain area. The sales volume in a certain month, or the sales volume of a certain product in all regions on a certain day, etc. In order to quickly respond to users’ query needs, basic data need to be calculated. The establishment of the cube is the result of a calculation, which can also be said to be the result returned by the query. Of course, this result may be an intermediate result or the final result. Whether it is an intermediate result or a final result depends on how the calculation matches the user’s actual query.

For an N-dimensional cube, the ith dimension level is $L_i$, the total number of cubes generated: $T = \prod_{i=1}^{n} (L_i + 1)$; that is to say, there are different query results. So is it necessary to calculate all $T = \prod_{i=1}^{n} (L_i + 1)$ possibilities? That is to say, a cube $T = \prod_{i=1}^{n} (L_i + 1)$? This involves the strategy of cube materialization. Completely materialized strategy, that is, it generates all $T = \prod_{i=1}^{n} (L_i + 1)$ cubes.

This strategy provides the shortest query response time. But the space cost is too high. This is a large storage space, long CPU inertia time, and the form of a computer cube if required quickly. In some implementation strategies, only part of the cube is generated. However, some implementations include cost evaluation and cube selection. This is a computing cube strategy limited by limited storage space and low demand requirements. The strategy of generating cubes is not implemented. This strategy sends quick answers to questions to a database system that stores raw data for processing without saving the realized cube. It is only suitable for situations with low demand and small scope. Figure 2 shows the network structure of MOLAP.

### 4 Massive data processing and multidimensional database management based on deep learning

#### 4.1 Mass data processing and multidimensional database performance testing based on deep learning

The analysis part of this article is mainly to test the performance and scalability of the test system. Two Inspur NP3060 devices are selected, and a high-memory, high-bandwidth network is selected. The installation system is VMware ESXi 5.5.0, and the virtual machine is constructed on this basis.

![Figure 2: MOLAP architecture.](image-url)
This part of the performance test uses the data as a virtual host and a virtual host. Each data collector simulates the temperature data of 1,000 points to 50,000 points in real time. The collection time is 1 s, the data copy number is defined as 2, and the test time is $7 \times 24$ h. The test results show that the system runs stably for 7–24 h. The test results are shown in Table 1.

From Table 1 and Figure 3, we can know that in the four-node-type tests, the highest CPU usage rate in the average range is 3.8%, the largest of the four highest value ranges is 12.1%, and the lowest is 0.2%. CPU usage varies greatly in different situations.

It can be seen from Table 2 and Figure 4 that the time used for sorting increases as the number of table records increases. When the number of table records is not too large, the sorting is relatively fast. The size of the buffer has a greater impact on the performance of the sort operation.

The speed of creating an index table is closely related to the size of the table. As the number of table records increases, the time it takes to create an index table also increases non-linearly. The size of the buffer has a greater impact on the speed of creating an index table.

### 4.2 Multidimensional database scalability and load balancing test

Use the data as a virtual central computer and a virtual central computer. Each data collector generates between 1,000 points and 50,000 points of temperature data

<table>
<thead>
<tr>
<th>Table 1: System performance test results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test conditions</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4 * 1,000</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4 * 5,000</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

*Figure 3: System performance test results.*
through real-time simulation. The collection time is 1 s, the number of copies is 2, and the test time is 7 × 24 h. The test results show that the system is stable within 7–24 h. The test results are shown in Table 3. Each column in the table is the number of data groups saved by the data node in the query. The test results show that the system can dynamically add and delete data nodes in real time and has high scalability. After a certain period of execution, the balance of data storage needs to be maintained. This shows that the system can balance the cargo.

### 4.3 Multidimensional database query test

One virtual machine is used as the main node, and four virtual machines are real-time data nodes, numbered from 1 to 4. The four virtual machines are data collectors and several clients. Each data collector generates 1,000-point temperature data through real-time simulation, the collection time is set to 1 s, and the number of copies is set to 3, which is 24 h of historical data. When the inferiority complex works normally and one or two real-time data nodes are disconnected, the customer will request historical data and record it in real-time data (including all

---

**Table 2: Sorting speed of the data list table and time to create an index table**

<table>
<thead>
<tr>
<th>Table records</th>
<th>Sorting in ascending order takes time (ms)</th>
<th>Sorting in descending order takes time (ms)</th>
<th>Index time (ms)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>148</td>
<td>154</td>
<td>411</td>
<td>The default number of buffer items in the buffer is 200</td>
</tr>
<tr>
<td>200</td>
<td>354</td>
<td>357</td>
<td>877</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>2,027</td>
<td>1,989</td>
<td>3,097</td>
<td></td>
</tr>
<tr>
<td>1,000</td>
<td>4,645</td>
<td>4,619</td>
<td>7,678</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3: Scalability and load balancing test results**

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,200</td>
<td>107,850</td>
<td>107,728</td>
<td>107,803</td>
<td>0</td>
</tr>
<tr>
<td>18,000</td>
<td>202,012</td>
<td>204,032</td>
<td>201,332</td>
<td>202,155</td>
</tr>
<tr>
<td>40,000</td>
<td>598,766</td>
<td>600,294</td>
<td>—</td>
<td>597,342</td>
</tr>
</tbody>
</table>

**Figure 4:** Sorting speed of the data list table and time to create an index table.
possible combinations). The test results of the past problem data are shown in Table 4. Allowed data are copied to a maximum of three nodes.

Figure 5 shows the results of the real-time recording test when the complex is normal. The real-time recording test result when the data node fails is shown in Figure 6. At time T1, real-time data nodes 2, 3, and 4 are shut down. At time T2, the real-time data node restarts, and the real-time data node 1 shuts down. At time T-3, the data node is reset to 3 in real time, and the data node is shut down to 2 in real time. Comparing Figures 5 and 6, we can see that when the data ratio is not large, only one real-time data node is in the complex memory, which will not affect data storage and subroutines. The test result shows that the data supported by the system are real time. When searching for historical data, the maximum number of data storage points in the system is the maximum number of data replication allowed by the system. The total load capacity of the node at the actual time of system existence is greater than the following data range. The system can guarantee data real-time processing.

### 4.4 Analysis of comparative results of multidimensional database experiments

This experiment will use the existing marketing system and the new statistical system to compare the time spent on statistical sales data and the accuracy of the data. The following comparison methods are specifically designed.

1. Statistical comparison of the sales volume of a certain product in the most recent period. Data query is performed daily during a week, and the time spent each time and the accuracy of the data are compared.
2. Statistical comparison of sales volume of a certain product over time. Perform data query for multiple time periods (period, month, and quarter), comparing the time spent each time and the accuracy of the data.

<table>
<thead>
<tr>
<th>Test premise</th>
<th>Number of clients</th>
<th>Test conditions</th>
<th>Accuracy (%)</th>
<th>Average query time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1h</td>
<td>1</td>
<td>Cluster health</td>
<td>100</td>
<td>0.0053</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect one</td>
<td>100</td>
<td>0.0055</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect two</td>
<td>100</td>
<td>0.0054</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>Cluster health</td>
<td>100</td>
<td>0.0616</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect one</td>
<td>100</td>
<td>0.0755</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect two</td>
<td>100</td>
<td>0.0842</td>
</tr>
<tr>
<td>Query 24h</td>
<td>1</td>
<td>Cluster health</td>
<td>100</td>
<td>0.0442</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect one</td>
<td>100</td>
<td>0.0586</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect two</td>
<td>100</td>
<td>0.0805</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>Cluster health</td>
<td>100</td>
<td>0.5542</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect one</td>
<td>100</td>
<td>0.7303</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disconnect two</td>
<td>100</td>
<td>0.9572</td>
</tr>
</tbody>
</table>
(1) Comparison of sales statistics of a certain product in the most recent period

Figures 7 and 8 show that the query time of the two systems is similar; the new system is only slightly less than the original system, but the accuracy rate is the same, and both are 100%. But because the data are updated in real time, the accuracy of the two systems is the same.

(2) Statistical comparison of sales volume of a certain product over time
Figures 9 and 10 show that the query time of the product sales volume of the two systems during the time period is much shorter than the original system, but the accuracy rate is somewhat lower. This is because the new system preprocessed the data from 1 month ago and saved it in the result set. When querying the sales volume for more than a month period, there is no need to query the database again, so the time is greatly reduced.
However, because the existing database operation is an update operation, if the order is updated more than 1 month later, the change data will be ignored. It is an exceptional individual situation. If the sales data are changed and additional methods such as cloud databases are used, the accuracy can be resolved.

4.5 Performance test of cluster data synchronization channel

The uploading data node is defined as the client, and the receiving data node is defined as the server. The server controls the output of the channel under the connection between 1 and 3 clients and the server. The test results are shown in Table 5.

5 Conclusion

This article mainly studies massive data processing and multidimensional database management based on deep learning, combined with the big data characteristics of the Internet of Things, the basic technology of big data processing, and the basic technical requirements of the Internet of Things; it focuses on the analysis and research of the massive data fusion algorithm based on deep learning. The model and the MOLAP model of the multidimensional database based on deep learning solve the problem of massive data processing in the Internet of Things.

The innovations of this paper are: firstly, this paper uses a combination of quantitative analysis and qualitative analysis; secondly, this paper uses a combination of empirical analysis and theoretical analysis; third, this paper fully integrates the deep learning model with multi-dimensional database management, improves data processing efficiency and data processing capabilities, and forms a massive data processing and multi-dimensional database management support system based on deep learning, which meets the massive data processing needs of the Internet of Things.

Multidimensional databases model data as factual, dimensional, or numerical measures, all of which perform interactive analysis of large amounts of data for making decisions. A multidimensional database is a leap in the field of data storage and provides a convenient platform for large-scale data storage. However, there are still some problems in using multidimensional databases. For example, to add a dimension to a multidimensional database, the database needs to be re-architected, and if the data have been stored before, it will fail to refresh, and a special way is needed to transform the dimension relationship of the data. The application prospects of the research on massive data processing and multidimensional database management based on deep learning proposed in this article are very broad.

Acknowledgment: This work was supported by the Analysis of Learning Characteristics of Online Learners in the Perspective of Smart Education, the Construction and Practice of Talent Cultivation Model for Innovative Experimental Classes of Information Majors in the Context of New Engineering, and a Study on the Innovative Reform of Undergraduate Programming Courses in Private Universities under the Background of OBE-Engineering Education Accreditation.

Funding information: This work is supported by the first-class undergraduate project "Linux Operating System" in Shaanxi Province, the Analysis of Learning Characteristics of Online Learners in the Perspective of Smart Education, the Construction and Practice of Talent Cultivation Model for Innovative Experimental Classes of Information Majors in the Context of New Engineering, and a Study on the Innovative Reform of Undergraduate Programming Courses in Private Universities under the Background of OBE-Engineering Education Accreditation.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

<table>
<thead>
<tr>
<th>Package size of each client (unit: KB)</th>
<th>1 Client (ms)</th>
<th>2 Clients (ms)</th>
<th>3 Clients (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,024</td>
<td>33</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>10,240</td>
<td>298</td>
<td>457</td>
<td>723</td>
</tr>
<tr>
<td>102,400</td>
<td>3,011</td>
<td>4,662</td>
<td>6,903</td>
</tr>
</tbody>
</table>
References


