Scholar Publishing Group
Machine Learning Theory and Practice
https://doi.org/10.38007/ML.2021.020203
ISSN 2790-0983 Vol. 2, Issue 2: 17-31

Massive Travel Data Based on Cloud Computing

Yanlin Ma, Wengu Ren and Jiahong Huang*

College of Economics and Management, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

1525267097@qq.com

**corresponding author*

Keywords: Cloud Computing, Big Data, Travel Data, Data Analysis

Abstract: With the rapid development of tourism and transportation, the explosive growth of travel information, travel data has formed a huge amount of information space. How to quickly, accurately and conveniently analyze the customer relationship of the massive amount of travel data that reflects the passenger information accumulated daily is of great significance for analyzing the operation status of the tourism market, predicting the impact of tourism on related industries, and adjusting the macro policy of tourism. This paper mainly studies the massive travel data based on cloud computing. This paper studies the mining and analysis of massive travel data, and proposes a parallel algorithm for travel data mining based on constraint association rules. The algorithm can solve the problem that the existing data mining analysis method is difficult to apply due to the long frequent itemsets of massive travel data. The algorithm can organize and mine various travel data according to the main characteristics of travel data, and can provide more valuable information for a specific region, a specific group of people or a specific need. The experimental data in this paper shows that when the number of concurrent requests increases, the response time of the system before and after the improvement is quite different. When the number of concurrent requests is 2000, the response time of the system is not much different. When the number of concurrent requests is 3000, the response time of the system is obviously different. When the number of concurrent requests is greater than 4000, the difference gradually becomes larger and larger, which indicates that the improved dynamic weighted round-robin algorithm is better than the original in terms of system response time.

1. Introduction

With the rapid development of tourism and transportation, the explosive growth of travel information, travel data has formed a huge information space [1]. How to quickly, accurately and

conveniently analyze the vast amount of historical data accumulated in daily travel that reflects travel information, and find the overall behavior of a specific region, a specific group of people or a specific demand in the disorderly data, such as entry and exit. Tourism, red tourism, eco-tourism, etc. are of great significance for analyzing the operation status of the tourism market, predicting the impact of tourism on related industries, and adjusting the macro policy of tourism [2-3]. Travel data has two basic characteristics: a large number of frequent item sets, and a common problem of how to organize the generation of things from a record set. For the two basic characteristics of data, in the existing constraint association rule mining algorithm, the separate algorithm has certain advantages for mining frequent item sets and relatively small lengths. The algorithm considers the filtering degree. The constraint is used to filter the database, and the candidate function set that satisfies the constraint condition is directly generated by the Join function in a low-upward manner [4-5]. However, when the frequent item set is long, the algorithm will generate a large number of candidate items, which makes it inefficient. When the amount of data is large, many algorithms for calculating frequent item sets cannot be processed or the calculation speed is too slow, which is difficult to apply in real-world scenarios [6-7]. For the problem of how to organize the generation of things, if you use the existing five constraints, you can't accurately solve the organization problem of travel data, and you can't greatly improve the efficiency of the algorithm of calculating association rules. On the one hand, you have to invest a lot of energy. The existing constraints are used repeatedly to meet the specific needs of travel data. On the other hand, the overall data mining algorithms are not well promoted, which affects the speed and effect of data mining and analysis [8-9]. With the popularization of information technology, the explosive growth of information content, and the corresponding improvement of information processing technology, social network data collection and analysis methods have been greatly improved, and large-scale quantitative analysis of social networks has become possible, especially in the tourism industry. The study of tourist relationship networks can reflect the dynamic trends of tourist hot spots in a timely manner [10-11].

Baek believed that cloud computing technology is a technology that provides computing resources on demand, so it can handle these challenges well because it has many excellent features such as energy saving, cost saving, agility, scalability and flexibility [12-13]. In his research, he proposed a cloud-based security framework for big data information management in smart grids, which he called "smart frameworks." The main idea of his framework is to build a hierarchy of cloud computing centers to provide different types of computing services for information management and big data analytics. In addition to this structural framework, he provides a security-based solution for identity-based encryption, signature and proxy re-encryption to address key security issues in the proposed framework [14-15]. Xin believed that with the development of technology, spatial data sets continue to grow at an alarming rate [16]. Traditional data management based on single-node DBMS can hardly meet the high concurrency requirements of massive data. The rise of cloud computing has brought new opportunities and challenges. Some researchers use a hybrid solution that combines fault tolerance, heterogeneous clustering, and distributed computing frameworks to achieve efficient performance. Shark is derived from Spark's computing framework and is a computational engine for fast data analysis. After submitting the query, Shark compiles the query into an operator tree represented by the RDD, which is then converted into a task map for execution by Spark. Shark does not currently support spatial queries; therefore, he introduces a way to enable Shark / Spark to support spatial queries. Using the API and UDF provided by Shark, Shark / Spark can process spatial data extracted from spatial databases and perform spatial queries based on requirements [17-18]. Wu believed that cloud computing offers the possibility to store and process large amounts of remotely sensed hyperspectral data in a distributed manner. Dimensionality reduction is an important task in hyperspectral imaging because hyperspectral data

often contains redundancy and can be deleted before analyzing the data in the repository. In this regard, the development of dimensionality reduction techniques in cloud computing environments can provide efficient storage and pre-processing of data [19]. In his research, he developed a parallel and distributed, widely used hyperspectral dimensionality reduction technique based on cloud computing architecture: principal component analysis (PCA). His implementation leverages Hadoop's Distributed File System (HDFS) for distributed storage, Apache Spark as the computing engine, and it is developed based on the map reduce parallel model. It makes full use of Hadoop's high-throughput access, high-performance distributed computing functions and cloud computing environment. He first optimized the traditional PCA algorithm, making it ideal for parallel and distributed computing, and then implementing it on a real cloud computing architecture. His experimental results using multiple hyperspectral datasets show that the proposed distributed parallel approach has high performance [20-21]. In view of the high real-time and computational accuracy, poor scalability of the platform, and low resource utilization requirements of power system simulation calculations, Li proposed a power system simulation cloud computing platform architecture based on Open Stack (open source infrastructure platform). He proposed a Hadoop (Parallel Processing Framework) that enables dynamic scaling, efficient computing and mass storage at low cost. Then, referring to the characteristics of power system simulation tasks, a virtual machine migration strategy based on multi-objective particle swarm optimization is proposed [22]. The (PSO) algorithm is used to implement resource scheduling of the cloud computing platform. During the virtual machine migration process, the hotspot is determined using an exponential smoothing based predictive model and the virtual machine is selected based on the migration speed and effect. At the same time, using multi-target PSO algorithm to search for nodes such as targ while ensuring the quality of power system simulation service, it also has the advantages of high resource utilization and low operating cost. Finally, simulation experiments were carried out on Cloud Sim, and the mobility and non-migration of the proposed algorithm and greedy algorithm were compared. His research results show that the proposed algorithm outperforms the other two algorithms in terms of service level agreement (SLA) violation rate, residual resource rate, energy consumption and number of virtual machine migrations. His experiments verify the advantages and feasibility of multi-objective PSO algorithm based on virtual machine migration in cloud computing platform resource scheduling [23-24].

The innovations of this paper are as follows:(1) Aiming at the shortcomings of weighted round-robin algorithm, a dynamic weighted round-robin algorithm is proposed. The algorithm utilizes the working state of the server, calculates the corresponding weights, assigns tasks, and improves the load effect. (2) For the C4.5 decision tree classification algorithm, the shortcomings of calculating the branching property are slow, and the problem of not being able to deal with large-scale data sets. Using Hadoop Map Reduce and Spark parallel computing framework respectively, this paper designs C4.5H and C4.5S in parallel. The algorithm scheme realizes the vertical division of the data set and the parallelization of the horizontal division of the branching attributes of the same layer of the decision tree, which improves the classification operation efficiency of the C4.5 algorithm. (3) Aiming at the shortcomings of KNN algorithm in finding K nearest neighbor distance and high redundancy, the KNN improved algorithm (FKNN) is presented in this paper. Based on the Spark parallel framework, the parallelization scheme of the improved KNN algorithm is implemented. Without the loss of classification accuracy, the classification efficiency of KNN algorithm in processing large-scale datasets is improved.

2. Proposed Method

2.1. Cloud Computing

2.1.1. The Concept of Cloud Computing

Cloud computing is a mature technology with a wide range of applications, and its definition is different. Wikipedia: Cloud computing integrates resources in the information technology field into services and provides users with the services they need through network connections. Cloud computing is based on distributed computing, virtualization technology, network computing and Web services. Users can access resources on demand and have unlimited access to large-scale data and problems at any time. Cloud computing uses virtualization and multi-tenancy technology. Virtualization is a resource processing technology. It abstracts and transforms various physical resources such as servers, computers, and networks. For the user, this is a huge resource integration. As a whole, it breaks the barriers that cannot be crossed between physical structures. Multi-tenant technology is a software architecture processing technology that guarantees the independence and isolation of user data. The premise is that multiple users use the same system or gradually use it. These two technologies simply summarize the sharing of physical resources for different users is virtualization technology; sharing the same software or component resources by different users is a multi-tenant technology. Currently, the business model of cloud computing is mainly composed of these two sharing technologies.

2.1.2. Cloud Computing Architecture

Cloud computing can link various resources, form a resource pool to provide users, realize the sharing of resources, and can easily implement expansion. Cloud computing consists of three layers: infrastructure layer, platform layer and application layer. In addition, there are management layers and security layers running through the three layers. The cloud computing architecture is shown in Figure 1.

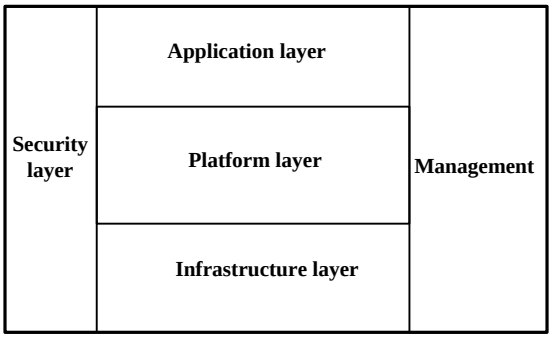

Figure 1. Cloud computing architecture

In the architecture of the cloud computing architecture:

1) The infrastructure layer, located at the bottom, mainly integrates servers, computers, storage and other devices to form resource pools for users to use. These resource pools can achieve unified management and provide basic support for the upper platform;

2) The platform layer, located above the infrastructure layer, provides an integrated environment for user service application deployment, and provides common conditions such as services, algorithm models, and operating environments, which can reduce the pressure on deployment on the business system;

3) The application layer, located at the top level, provides services for users to meet the needs of users' business use;

4) Management and security layer, unified coordination, management and monitoring of infrastructure, platform and application layers.

2.1.3. Characteristics of Cloud Computing

The scale is huge: it usually consists of tens of thousands of units and hundreds of thousands of services. The scale is very large. Virtualization: Users do not need to care about the specific location. The services can be obtained through any location and various terminals. The resources are uniformly distributed by the system. For the user, all the requested resources come from the cloud, not the specific server; the reliability is high: the cloud computing has multiple copies, and the computing nodes can be easily interchanged. These technologies can ensure data redundancy and reliability; versatility good: The same cloud can support different applications at the same time, not for specific applications; high scalability: the cloud is integrated by physical servers, computers, storage and other devices, and achieve unified management, this mode is easy to expand. It can easily meet the user's growth demand; the cost is low: the cost of a single cloud node is very low, and the computing power and storage capacity are very strong when integrated into the resource pool. For the user, it is only necessary to care about the service that they want to get, and there is no need to maintain physical equipment, which saves a lot of maintenance costs. The resource pool is managed centrally and automatically by professionals, which in turn can save a lot of costs, so as to support users to provide affordable services.

2.1.4 Classification of Cloud Computing

According to the scope of deployment, cloud computing can be divided into three categories: public cloud, private cloud and hybrid cloud.

1). A private cloud, also known as a proprietary cloud, is generally deployed within a company or organization and provides services only within the enterprise or organization and is not open to the public. Private clouds are flexible, easy to maintain and manage, and have great privacy.

2).the public cloud, the public cloud can provide services for organizations, enterprises or individuals in need, generally provided by a dedicated cloud service provider, the cloud server provider deploys the infrastructure itself, the user only need to request services through their own terminal devices, the user generally has to pay a certain fee for the service.

3). hybrid cloud, hybrid cloud is a model that combines public and private clouds. Using a hybrid cloud can take advantage of the public and private clouds. For services that require high security and high security requirements, they are deployed on the private cloud. For general services, services that are not confidential and whose resources are not enough can be deployed on the public cloud. Hybrid cloud deployment can effectively shorten the construction period and save construction costs, and can meet both security requirements and convenience requirements.

2.2 Data Mining Theory

Data mining, also known as Knowledge Discover in Database (KDD), is a process of discovering useful knowledge from a large number of incomplete fuzzy application data in a database. It is an interdisciplinary subject in the fields of statistics, databases, machine learning, artificial intelligence, high performance computing, and data visualization. This picture says that data mining is like finding gold from sand and stone, and finding small and valuable parts from a lot of raw materials. The term "data mining" is named after it.

2.2.1. Basic Steps of Data Mining

In general, a complete data mining process involves the following steps:

Data cleansing (removing noise and data inconsistency): There are more or less inconsistent data records in the data set in the database. Data mining algorithms cannot be used directly for unqualified data sets. Data cleansing can improve the quality of data records by filling in missing data values, smoothing data noise, and eliminating data outliers to meet the specifications and requirements of mining algorithms.

2) Data integration (combined with multiple data sources): Combine data from multiple data sources to form a consistent data store, sometimes need to clean up the data to eliminate possible data redundancy.

3) The data protocol (extracting data related to the analysis task from the database): Under the premise of not affecting the data mining results, the data set size is compressed by data aggregation and deletion of redundant features, and only data mining is retained. Reduced the time complexity of data mining.

4) Data conversion (transform data into a form suitable for mining): There are many data conversion methods, including smoothing, aggregation processing, normalization, data generalization processing, attribute construction, and so on. In addition, if the data is real, you can also use the concept of layering and data discretization to transform the data.

5)Knowledge Discovery (Using Algorithms to Extract Useful Knowledge in Data Sets): Knowledge discovery is one of the core steps of data mining. It uses data mining algorithms to analyze data sets in data warehouses to find useful data patterns.

6)Model evaluation (evaluation of knowledge discovery results according to certain measurement methods): Removing patterns that do not meet the evaluation criteria often requires a series of objective evaluation criteria, such as the accuracy, support, confidence, and validity of the rules. Verify the correctness of the data mining results from a practical perspective.

7) Knowledge representation (displaying the extracted knowledge in a visual way): Visual analysis can be used to visually display the analysis results obtained by data mining to the user. Of course, the results of the analysis can be stored in the database for other applications to call. The data mining process is often not completed at one time, it is a cyclical process. If a step does not meet the expected goals, you will need to adjust and re-execute the process. The data mining steps from step 1 to step 4 can be summarized as data preprocessing. Simply put, the steps of data mining include: data preprocessing, knowledge discovery, pattern evaluation, and knowledge representation. The specific steps are shown in Figure 2.

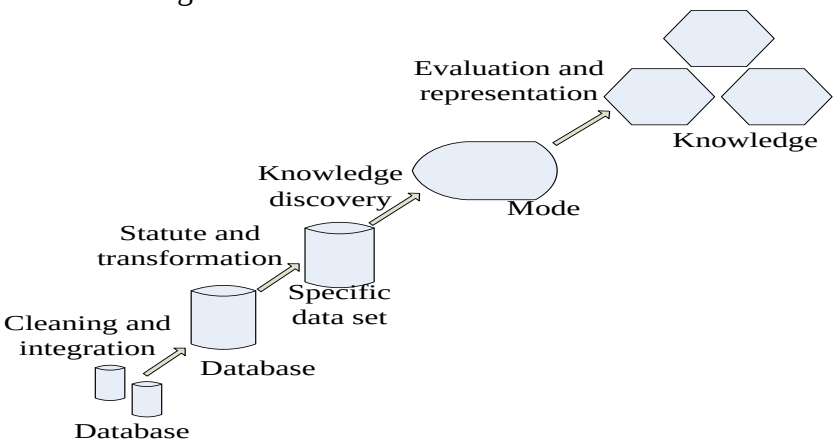

Figure 2. Schematic diagram of the data mining process

2.2.2. Association Rules Mining Process

In general, the mining of association rules can be seen as a two-step process:

l) Find all frequent item sets: By definition, each frequent item set appears at least as often as the predefined minimum support count min_sup.

2) Frequent item sets produce strong association rules: by definition, these rules must meet minimum expenditure and minimum confidence. Once a transaction in database D discovers a frequent item set, it can generate powerful association rules directly from it. For confidence, it can be calculated using the following formula:

$$confidence\,(A \Rightarrow B) = P(A|B) = \frac{\sup port\_count(A \cup B)}{\sup port\_count(A)} \tag{1}$$

The conditional probability is represented by the number of support for the item set, where the support number (A UB) is the number of transactions with the item set AUB, and the support number C(A) is the number of transactions containing the item set A.

2.2.3. Rule Metrics

The support and confidence of the rules are two important indicators for measuring the benefits of the rules. They reflect the usefulness and certainty of the rules found. Supporting a 0.5% association rule means that 0.5% of all analyzed transactions buy diapers and beer at the same time. A 60% confidence level means that 60% of diaper customers will also buy beer. In general, the association rule is very interesting if the association rule satisfies both the minimum supported flash value and the minimum confidence threshold. These thresholds can be set by the user or expert. Other analyses can also reveal interesting statistical correlations between related projects.

Let $I = \{I1, I2, ..., Im\}$ be a collection of projects. Let task-related data D be a collection of database transactions, where each transaction T is a collection of items, $T \in I$, and each transaction has an identifier called a TID. Let A be the project set, and transaction T contains A if and only if A $\in$T. The association rule is a corollary of A=>B, where A$\in$I, B$\in$I, and $A \cap B = \varnothing$. Rule A => B is established in transaction set D supporting S, where S is the percentage of transactions in A containing AUB. It is the probability P (AUB) rule A => B has a confidence level c in the transaction set D, where c is the percentage of the transaction D containing A also containing B, which is the conditional probability P (B|A). this is

$$\sup port(A \Rightarrow B) = P(A \cup B) \tag{2}$$

$$confidence\,(A \Rightarrow B) = P(B|A) \tag{3}$$

A general association rule is considered to be meaningful when it meets the minimum support threshold (min sup) and the minimum confidence threshold (min conf), called a strong rule.

2.3. Hadoop Theory

2.3.1. Density Peak Clustering Algorithm

The idea of this algorithm is relatively new. The idea of the algorithm center is based on the following assumptions: First, the density center point of the cluster (the point with the highest density in the cluster) is not lower than the point near the cluster. Second, the density center point of the cluster is usually far from the point where the density of the data set is high (this usually belongs

to another cluster). In the algorithm idea of the density peak clustering algorithm, there are two attributes for each point: (1) The density value ρ; (2) The repulsion value δ (the minimum value of the distance from the point where the density value is larger than itself). The larger the density value ρ is, the more likely the point is the density center of the cluster. The larger the rep group δ is, the more likely the point is to represent a new cluster, because when a point is denser than the point, the distance is higher. At very far, the point in the space between the point and the point where the density is larger is denser than the point in the vicinity of the point. In spatial data distribution, clusters and clusters often have low-density regions to distinguish between them. Therefore, the larger the repulsion value δ indicates that the point is more likely to represent a new cluster. Therefore, it can be obtained that only when the density value ρ and the repulsion value δ are large, the point may be the density center point of a certain cluster. Considering the data set to be clustered, for any data point x_i in S, the density peak clustering algorithm calculates its local density ρ_i and the cluster value δ_i.

The local density ρ_i calculation formula is:

$$\rho_i = \sum_j x(d_{ij} - d_c) \tag{4}$$

In formula (4)

$$x(x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases} \tag{5}$$

The parameter dc> is the cutoff distance, dc is the user-defined distance threshold, and the literature gives a more appropriate empirical value, that is, the distance between all pairs of points in the data set is sorted from large to small. The value at %-2%. For example, 100 points, a total of 3950 combinations, sort these combined distances, select a combined distance of 50-100 as the dc value. As can be seen from equation (4), ρ_i indicates the number of S midpoints whose distance x_i is smaller than dc (regardless of x_i itself). The δ_i of point i is defined as:

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij}) \tag{6}$$

The density value of the minimum point of the distance represented by the repulsion value δ is larger than itself. Suppose the density is greater than ρ_i, point j is closest to i point, then $\delta_i = d_{ij}$, point j is the point where point i is connected to σ_i and $\sigma_i = \arg\min_{j \in s, \rho j > \rho i}(d_{ij})$ points. Point i can be attributed to the cluster to which point j belongs. The smaller the δ_i group value is, the more likely it is to attach. It is explained that point i is more likely to belong to the cluster to which point j belongs. The larger the copy value, the farther i point is from j point, and the weaker the dependency. More likely, point i and point j do not belong to the same cluster, that is, outliers. When the density value ρ_m of the point m is the maximum density of all points, the rep group value δ_m of the point m is

$$\delta_m = \max_j(d_{mj}) \tag{7}$$

When calculating the density value ρ of each point, since the data in this group may require data

in other groups, the EDDPC algorithm performs inter-group copying of the data according to the following formula.

$$\frac{|o, p_i|^2 - |o, p_j|^2}{2 \times |p_i, p_j|} < d_c \tag{8}$$

Where pi, pj are the seed points of the group p_i, p_j, so that the points of the adjacent group in the range of the boundary line dc can be copied to each other, so that the correct density value ρ can be obtained when calculating in the group. However, the copying method may cause the problem shown in Figure 3. The area where o2 is located is redundant, and because of the redundant copying, the calculation of the density value ρ of each point in the group also causes redundant calculation.

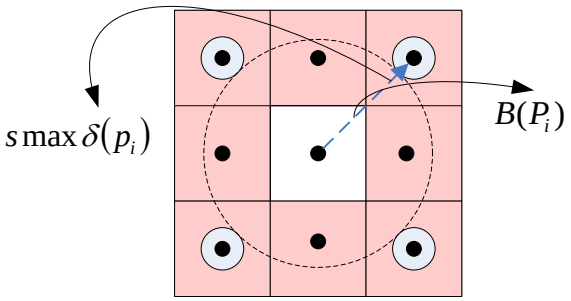

Figure 3. Example of EDDPC algorithm δ' replication method

Similarly, in calculating the repulsion value δ, in order to accurately calculate the δ value of each point, it is necessary to copy the data again according to the following formula.

$$|o, p_i| \le B(P_I) + s\max\delta(P_i), \rho_o > \min\rho(P_i) \tag{9}$$

$$|o, p_i| \le \min\{2|m, p_i| + |p_j, u| + |p_i, p_j|\}, \rho_o > \max\rho(P_i) \tag{10}$$

Among them $B(P_i) = \max\{|o, p_i|, \forall_o \in p_i\}$, $s\max\delta(P_i)$ is the second largest δ in the group, m is the most dense point in the group P_i, $u \in P_j$.

3. Experiments

3.1. Experimental Design and Experimental Equipment

In this paper, our experimental environment is a cluster system consisting of 7 servers. The servers are all HP blade servers, configured with Intel x86 and 4G memory; the hadoop version used is 0.19.0.

Based on the HDFS architecture, this paper adds independent small file storage modules, by merging small files, building indexes, and increasing cache, we can effectively store large amounts of small file data, reduce the memory consumption of metadata nodes, and improve the search efficiency. When reading or writing a file, you must first determine whether the file belongs to a small file. If it is not a small file, you do not need to process it and directly hand it to HDFS for processing. If it is a small file, it is handed to the small file storage module for processing.

3.2. Data Collection

The data of the experiment in this paper is part of the passenger information of an airline. The specific record format is <FLIGHT, OFFDAY, ARRDAY, PAI, HSET, ENGLISHNAME, CHINESENAME, FOID, CLASS, SEAI, PNR, ID>, where FLIGHT is the flight number. OFFDAY is the departure date, ARRDAY is the arrival date, STRT is the departure place, PATHSET is the path (this data is STRT and DEST, STRT is the departure place, DEST is the arrival place), ENGLISHNAME is the passenger English name, CHINESENAME is the passenger Chinese name, FOID For passenger ID number, CLASS is the aircraft cabin number (such as F, G is first class, normal class), SEAT is the seat number, PNR is the unique number of passenger booking information, ID is the record number.

4. Discussion

4.1. Cloud Computing Based Massive Travel Data Analysis

4.1.1. Comparison of Dynamic Weighted Round Robin Algorithm and Unconstrained Parallel Apriori Algorithm Running Time and Number of Association Rules

Table 1. Algorithm runtime chart

	2%	3%	5%
Relational extended path constraint algorithm	2235	1978	6982
Unconstrained parallel Apriori algorithm	6983	4103	8621

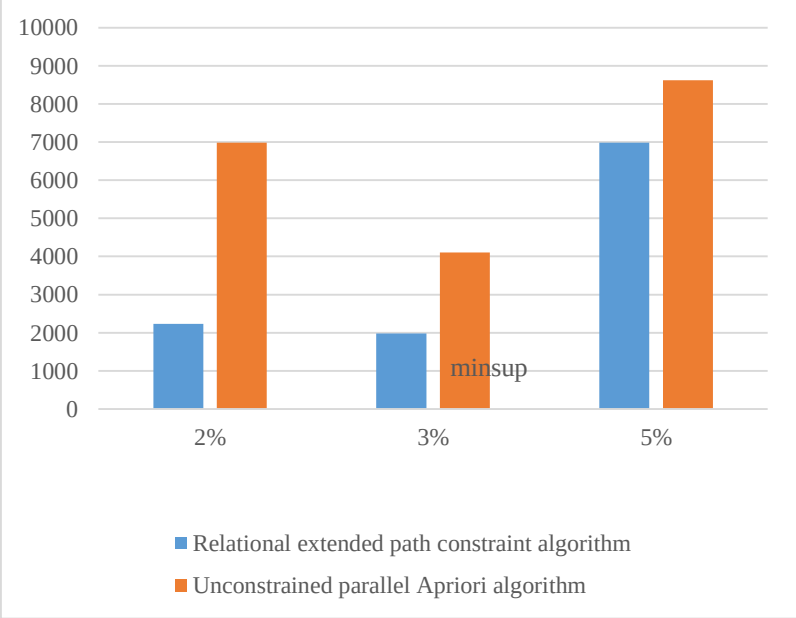

Figure 4. Algorithm running time comparison chart

As shown in Table 1, statistical analysis was performed for the experimental results. Figure 4 is obtained from experimental data analysis. As shown in Figure 4, after comparing the running time of this algorithm with the unconstrained parallel algorithm, when minsup is 2%, the speed is increased by 67.7%. When minsup is 3%, the speed is increased by 116.9%. When minsup is 5%,

the speed is increased by 210.4%. In general, the algorithm is effective, especially in the processing of massive data. It has obvious advantages. The mining speed is about 2.5 times that of the unconstrained parallel Apriori algorithm, and the number of generation rules is reduced by about 50%. The rules of meaning make the results more close to the actual situation and more instructive.

4.1.2. Comparative Analysis of Execution Time under Different Data Volumes

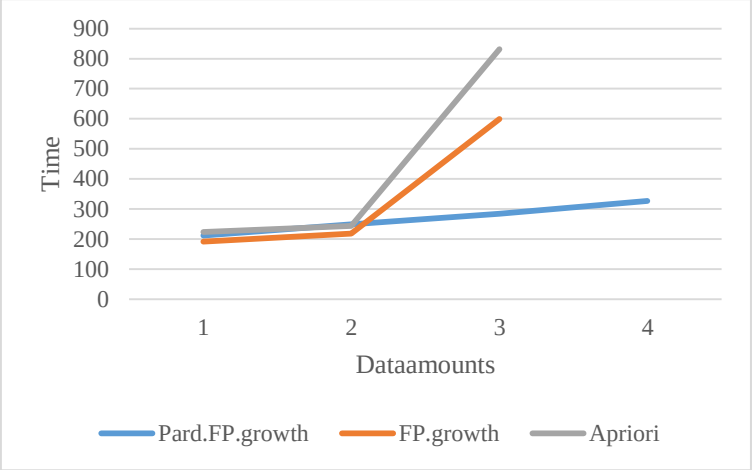

Figure 5. Execution time for different data volumes

As shown in Figure 5, the execution time comparison of the three algorithms in the same minimum support (0.3) for different data volumes (30M, 60M, 90M, 120M); when the amount of data is small, the advantages of the parallel FP algorithm are not obvious. When the amount of data is large, the execution time of the FP-Growth algorithm and the Apriori algorithm is greatly increased. This is because the FP-Growth algorithm only needs to scan the database twice to generate all the frequent item sets. Most of the work is done. It is done in memory. When the amount of data is large enough, the single CPU system will not be able to perform association analysis; the Apriori algorithm needs to generate a large number of candidate sets, so the execution time of the algorithm is relatively longer. The time of the system decreases as the minimum support increases. Because the higher the minimum support, the more frequent item sets are eliminated, making the time overhead for processing the frequent pattern tree and scanning the database less.

4.1.3. Comparative Analysis of Memory Consumption of Metadata Nodes

In order to verify the effectiveness of the scheme, this paper selects different types of small files for experiments. The file types include txt files, jpg files, jpeg files, png files, pdf files, dat files, rar files, and so on. Five sets of experiments were performed on the data, and the average was taken as the experimental result. The experimental file size and quantity are shown in Table 2.

Table 2. Test data sheet

File	The first group	The second group	The third group	The fourth group	The fifth group	The sixth group
The file size	500KB	200KB	100KB	50KB	20KB	50KB
The number of files	100	400	1000	5000	8000	10000

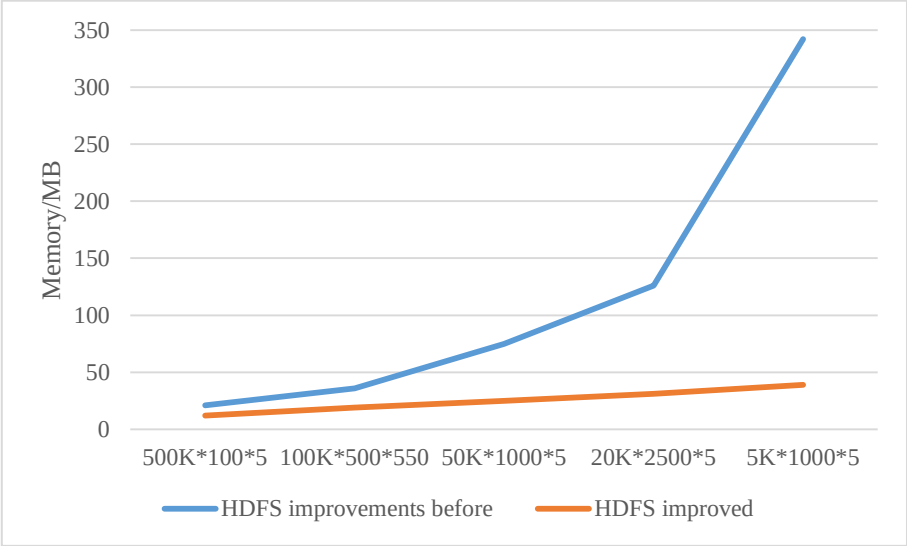

Figure 6. Metadata node memory consumption comparison chart

As can be seen from Figure 6, when the number of small files is small, the memory consumption of the metadata nodes before and after the HDFS improvement is not much different. When the number of small files increases, the memory consumption of the metadata nodes before and after the HDFS improvement increases. In contrast, HDFS before the improvement does not process small files, and increases dramatically as the amount of small files increases. The improved HDFS has a series of operations such as merging small files, indexing, and adding caches, and the number of small files increases slowly as the amount of small files increases. Explain that the improved HDFS can alleviate the problem of massive memory consumption of metadata nodes.

4.1.4. Comparative Analysis of Algorithm Execution Efficiency

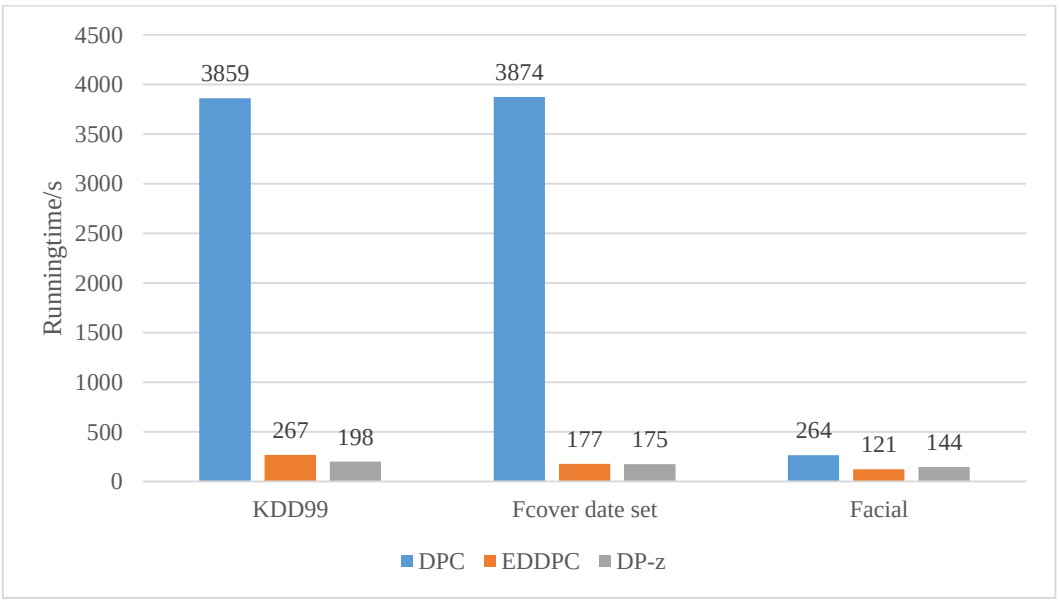

Figure 7. Comparison of algorithm runtimes

As shown in FIG. 7, the DP-z, EDDPC, and DPC algorithms of the comparison algorithm operate on the KDD '99_10% of the first 100k data set and the F Cover Type data set. The results

show that the algorithm is significantly better than the original algorithm. During the implementation, both DP-z and EDDPC algorithms use filtering measures to reduce a large amount of redundant calculations. Comparing the DP-z algorithm with the EDDPC algorithm, it can be seen that the execution time of the DP-z algorithm on the KDD '99_10% and F Cover Type data sets is shorter than the EDDPC algorithm, and the EDDPC algorithm performs shorter on the face data set than the EDDPC algorithm. In DP-z, it is found that the data amount of the data set contained in the face is small, the time used by DP-z and the EDDPC are basically the same distance calculation, and the DP-z algorithm needs to calculate the z value of the data point, so the amount of data of additional overhead for KDD'99_10% and F Cover Type is very large, and the data point z value needs to be calculated. However, since the EDDPC algorithm contains a large number of redundant distance calculations, the overall runtime is longer than DP-z. It can be seen that compared with the EDDPC algorithm, the DP-z algorithm studied in this paper has larger data volume and more obvious advantages.

5. Conclusion

This paper studies load balancing and improves the weighted round-robin algorithm. The weight is calculated dynamically according to the running state of the server. After the improvement of the weighted round-robin algorithm, the algorithm has improved in response time, throughput and actual concurrency. Then based on the theoretical research, through the load test tool LoadRunner simulation experiment, it is verified that the improved weighted round-robin algorithm can improve the performance of the algorithm to a certain extent.

The main work of this paper includes the following aspects: (1) Verify the performance of the algorithm through simulation experiments. The experimental results show that the improved algorithm can accurately evaluate the working state of the server, obtain reasonable weights, and verify the effectiveness of the algorithm. (2) For the problem of massive small file data storage, a solution for adding small file storage modules based on HDFS is proposed. The program is divided into small files for merging, indexing and caching mechanisms. It can satisfy the storage of large amount of small file data and has certain scalability.

For the weighted round robin algorithm, a dynamic weighted rounding algorithm is proposed. After trial and analysis, it can improve the effect of load balancing to a certain extent, but there are still some areas to be improved, mainly in the following two aspects: (1) The load balancing algorithm depends to a large extent on the accurate evaluation of the server's operating conditions. Thereby guiding the assignment of requests. The periodic collection of the server's operating status can accurately assess the state of the server to a certain extent, but it requires a large amount of resources and cannot be updated at the right time. You can consider the automatic feedback of the server, feedback the change of the running state according to the server's own situation, which can improve the accurate evaluation of the server and improve the performance of the algorithm. (2) The experiment is carried out in a simulated environment. It does not combine real application scenarios and should be tested and applied in the actual application environment.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Baek J , Vu Q H , Liu J K , et al. A Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid. IEEE Transactions on Cloud Computing, 2015, 3(2):233-244, (Doi: https://doi.org/10.1109/TCC.2014.2359460).

[2] Xiaoyue L , Qiang G . Cloud computing based cluster analysis on data of power utilization. Journal of Liaoning Technical University(Natural Science), 2016, 21(11):2223-2226.

[3] Wang R D, Sun X S, Yang X, et al. Cloud Computing and Extreme Learning Machine for a Distributed Energy Consumption Forecasting in Equipment-Manufacturing Enterprises. Cybernetics and Information Technologies, 2016, 16(6):83-97 (Doi: https://doi.org/10.1515/cait-2016-0079).

[4] Helmymohamed A , Youssif A A A , Ghalwash A Z . Cloud Computing Security Framework based on Elliptical Curve. International Journal of Computer Applications, 2015, 110(15):45-51 (Doi: https://doi.org/10.5120/19395-1069).

[5] He Q , Zhao B , Chang L , et al. PSSRC: A Web Service Registration Cloud Based on Structured P2P and Semantics. International Journal of Data Warehousing & Mining, 2016, 12(2):21-38 (Doi: https://doi.org/10.4018/IJDWM.2016040102).

[6] Petri I, Li H, Rezgui Y, et al. A HPC based cloud model for real-time energy optimisation. Enterprise Information Systems, 2016, 10(1):108-128 (Doi: https://doi.org/10.1080/17517575.2014.919053).

[7] Yang S , Qiu Y , Shi B . The Key Technology Study on Cloud Computing Platform for ECG Monitoring Based on Regional Internet of Things. Zhongguo yi liao qi xie za zhi = Chinese journal of medical instrumentation, 2016, 40(5):341-343.

[8] Liu X, Guo Q. Cloud computing based cluster analysis on data of power utilization. Journal of Liaoning Technical University, 2016, 21(11):2223-2226.

[9] Yoshinobu T , Shigeru Y . Reliability Analysis Based on a Jump Diffusion Model with Two Wiener Processes for Cloud Computing with Big Data. Entropy, 2015, 17(12):4533-4546 (Doi: https://doi.org/10.3390/e17074533).

[10] Chen G , Wang E , Sun X , et al. An Intelligent Approval System for City Construction based on Cloud Computing and Big Data. International Journal of Grid and High Performance Computing, 2016, 8(3):57-69 (Doi: https://doi.org/10.4018/IJGHPC.2016070104).

[11] Zhao Z , Li J , Liang J H , et al. An Empirical Study of Medical Big Data and Hospital Information System Construction Based on Cloud Computing. Journal of Computational and Theoretical Nanoscience, 2016, 13(12):10358-10363 (Doi: https://doi.org/10.1166/jctn.2016.6165).

[12] Ogiela, L. , Ogiela, M. R. , & Ko, H. . Intelligent data management and security in cloud computing. Sensors, 2020, 20(12): 3458 (Doi: https://doi.org/10.3390/s20123458).

[13] Mahmoud Ismail , Naif El-Rashidy , Nabil Moustafa, Mobile Cloud Database Security: Problems and Solutions, Fusion: Practice and Applications, 2021, 7(1): 15-29 (Doi: https://doi.org/10.54216/FPA.070102).

[14] Baek J , Vu Q H , Liu J K , et al. A Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid. IEEE Transactions on Cloud Computing, 2015, 3(2):233-244 (Doi: https://doi.org/10.1109/TCC.2014.2359460).

[15] Zhang Z , Li F . A Dynamic Management Method of Domestic Internet of Things Based on Cloud Computing Architecture. Journal of Computational & Theoretical Nanoscience, 2016,

13(12): 9963-9967 (Doi: https://doi.org/10.1166/jctn.2016.6095).

[16] Hisham Elhoseny , Hazem EL-Bakry, Utilizing Service Oriented Architecture (SOA) in IoT Smart Applications, Journal of Cybersecurity and Information Management, 2019, 0(1): 15-31 (Doi: https://doi.org/10.54216/JCIM.000102).

[17] Xin W , Kan L , Rongguo C . A Framework of Distributed Spatial Data Analysis Based on Shark/Spark. Journal of Geo-Information Science, 2015, 17(4):401-407.

[18] Wang Z J, Mujib A B M M. The Weather Forecast Using Data Mining Research Based on Cloud Computing.. Journal of Physics Conference, 2017, 910(1):012020 (Doi: https://doi.org/10.1088/1742-6596/910/1/012020).

[19] Vinay Padimi, Venkata Sravan Telu, Devarani Devi Ningombam, Applying Machine Learning Techniques to Maximize the Performance of Loan Default Prediction, Journal of Neutrosophic and Fuzzy Systems, 2022, 2(2): 44-56 (Doi: https://doi.org/10.54216/JNFS.020204).

[20] Wu Z, Li Y, Plaza A, et al. Parallel and Distributed Dimensionality Reduction of Hyperspectral Data on Cloud Computing Architectures. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 2017, 9(6):2270-2278 (Doi: https://doi.org/10.1109/JSTARS.2016.2542193).

[21] Kai P, Leung V C M, Huang Q. Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System Over Big Data. IEEE Access, 2018, 6(99):11897-11906 (Doi: https://doi.org/10.1109/ACCESS.2018.2810267).

[22] Rajalakshmi, M., Saravanan, V., Arunprasad, V., A., C., Khalaf, O. I. et al. Machine Learning for Modeling and Control of Industrial Clarifier Process. Intelligent Automation & Soft Computing, 2022, 32(1), 339-359 (Doi: https://doi.org/10.32604/iasc.2022.021696).

[23] Li Z , Zhu S , Hong H , et al. City digital pulse: a cloud based heterogeneous data analysis platform. Multimedia Tools and Applications, 2017, 76(8):10893-10916 (Doi: https://doi.org/10.1007/s11042-016-4038-2).

[24] Feng Z, Hang Z, Hai J, et al. A Skip-gram-based Framework to Extract Knowledge from Chinese Reviews in Cloud Environment. Mobile Networks & Applications, 2015, 20(3):363-369 (Doi: https://doi.org/10.1007/s11036-015-0612-5).