ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

Distributed Data Consistency Algorithm in Database Based on Language Recognition

Ting Chen[1,a,*], Haiyun Wang[1,b]

[1]*Xi'an Siyuan University, Xi'an, Shaanxi, 710038, China*
[a]*37790356@qq.com,* [b]*906138020@qq.com*
[*]*Corresponding author*

Abstract: *With the development and popularization of computer technology, people's dependence on computers is becoming stronger and their requirements for computers are also increasing. Traditional independent computer systems can no longer meet people's needs in terms of functionality and performance. Distributed systems are systems where multiple computer systems are connected through a network and are becoming increasingly common. In the field of databases, distributed database systems have become an important research field. Distributed database systems are becoming increasingly popular because they meet the needs of current information system applications and meet the popular management ideas and methods in today's business organizations. At the same time, the industry has also begun to actively research and develop distributed database systems. In distributed databases, data redundancy is achieved by copying the same data block to multiple nodes for application and security reasons. If the database platform of a distributed system is not necessarily the same on all nodes, the problem to be solved is how to maintain data consistency with minimal duplication throughout the entire database system.*

Keywords: *Database, Distributed Data, Consistency Algorithm, Language Recognition*

1. Introduction

Distributed database system is a product that combines distributed computing, processing, and network technology. It not only has strong management capabilities for dispersed data, but also has good performance. With the expansion of database applications, distributed database systems are receiving increasing attention and are currently one of the most popular research directions. A distributed database is a database that physically disperses many data sources on a computer network. Due to the distribution of data in distributed databases across multiple nodes, there is a significant difference in the integrity of their data compared to centralized databases.

Scholars have conducted relevant research on the issue of distributed data consistency in databases. Sattler F proposed a new multi-objective collaborative learning system, which uses the geometric characteristics of FL (filrate loss) loss surface to divide customer groups into multiple clusters by sharing training data. Compared with existing FMTL (Functional Munitions Task Listing) methods, CFL (Custom Function Library) does not require modifications to the FL communication protocol and is generally suitable for unconventional applications (especially deep neural networks). It does not require prior knowledge of the number of clusters and provides strong mathematical guarantees for the quality of clustering. Since clustering only occurs after FL converges to a stationary point, CFL can be considered as a post-processing method that always achieves better or equivalent performance than traditional FL and allows customers to obtain more professional models [1]. Lu Y has developed a distributed architecture for secure multi-party data sharing based on blockchain. In the rest of this paper, data sharing is conceptualized as a machine learning problem, involving collaborative learning and privacy protection. Shared data models can protect privacy rather than leaking actual data. Finally, the experiment combines collaborative learning with the consensus process supporting the blockchain, so that the computing activities used for consensus can be used for collaborative learning. The numerical results on real datasets show that the proposed data sharing scheme achieves good accuracy, high throughput, and high security [2]. Kaur H believes that in practical applications, the imbalance of data is a huge challenge. In computer vision, information security, marketing, medicine, and other fields, people often find that the original raw materials are often distorted by the distribution of a set of data. This project aims to compare and analyze existing standard methods, algorithms, and biased data

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

analysis methods, as well as the similarities and differences of various methods in different data distribution and application fields [3]. The application scope of distributed data consistency algorithms in the above research databases is relatively narrow.

In the process of data processing, data quality is a crucial issue. Inconsistency testing is an important aspect of data quality research. Although inconsistent data is widely present, it is not obvious. Utilizing these inconsistent data not only leads to decision-making errors, but also causes significant losses. In real life, in order to improve data quality, it is often necessary to find inconsistent data, that is, data inconsistency testing. For a centralized database, inconsistency testing is relatively simple. For example, for functional dependency testing of multiple data, a SQL (Structured Query Language) based testing method can be used. However, in real life, data from multiple relational tables is often divided into multiple parts, and data from multiple parts may be split. This paper first introduces the distributed data and consistency algorithm clearly, and then verifies the progressiveness of the algorithm through experimental analysis, which is of research significance.

2. Distributed Data and Consistency Algorithms for Language Recognition

2.1 Distributed Data Consistency Algorithm

Data replication is a commonly used method in applications such as distributed databases and data warehouses. However, during the data replication process, two types of data updates occur: synchronous and asynchronous. During the transaction execution process, the Synchronous copy method allocates updates. After the transaction is completed, each update would be simultaneously transmitted to other nodes of the transaction (using the locking protocol to obtain the corresponding lock), and the transaction would be simultaneously transmitted to all transaction copies. Synchronous replication also increases the cost of reverse transactions, as all nodes, including replicators, are informed that they must stop updating reverse transactions. The asynchronous replication mode also has its drawbacks, with the most serious being the problem of recommending stale data [4].

When a person emits sound, it is received through a microphone and then converted into an analog-to-digital converter chip to obtain a one-dimensional speech signal. However, one-dimensional time-series signals are not suitable for speech recognition due to their large data volume and unclear features. Therefore, they are generally mapped to frequency through Fourier transform (FFT) to obtain two-dimensional frequency signals as input for speech recognition systems. Common voice signals include: FFT (Fast Fourier Transform), Fbank (filterbank), MFCC (Mel Frequency Chemical Coefficients), PCEN (Per channel energy normalization), etc. Fbank features and MFCC are analog to the human ear auditory system, which suppress high-frequency signals and reduce feature dimensions, while PCEN features introduce normalization

The most complex query in a distributed database system is a federated query, which involves querying data from multiple objects, creating remote tables, and moving, merging, and combining them [5]. If a retail chain store in Shanghai wants to query the inventory data of all books sent by suppliers in Beijing to neighboring retail chains in Nanjing or Suzhou, it first obtains remote objects from the parent company and the network in Nanjing/Suzhou. Then the parent company creates a temporary data table TL using external objects, purchases all supplier data from Table T1 for Beijing (the store only has supplier names, no location data, only the parent company has supplier data), and then purchases TL tables for Nanjing and Suzhou. These stores are stored in chain stores. Then, it merges the T1 table with the inventory tables of these two stores, finds all the book data provided by the Beijing supplier (titles, quantities, copies, prices, etc.), and sends the merged results to the Shanghai store chain and T3 data table. Finally, use the Datagrid controller to merge and display T2 and T3 tables. After copying, moving, and merging data tables and inserting them into a DLL (DynamicLinkLibrary), including other pages for each function (remote creation of data tables, remote merging, and inter table merging, etc.), the Datagrid controller can create data tables [6].

If an object data event contains more than one change, it consists of more than one sub event. A transaction consists of two main execution stages, namely the preparation stage and the commitment stage. The preparation stage includes preparation work and answering questions, while the commitment stage includes the final determination of all sub transactions. The result of these two stages is that the first stage leads to the delivery of all sub transactions, and the second stage leads to the return of all sub transactions, allowing different parts to be fully implemented and postponed. Therefore, the two-stage transmission method mainly provides data consistency through integrity. This method allows

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

applications to perform coordinated update operations on two or more servers, so two independent transactions can be considered as one. A commit server can record all commit and rollback decisions for a transaction, but only the central registration authority can do so. Using the two-step commit method can ensure that the database updates on the server are consistent, either all or none.

In the two-step TRANSACTION method, the user sends the following sequence during the preparation phase: BEGINTRANSACTION transaction to SQL statement, and PREPARE TRANSACTION0N to conversion statement. The processes involved must execute the operations in this sequence when participating. The former is used to identify application processes, transfer service processes, and transactions, while the latter is used to modify content when statements indicate readiness for transfer. Once the preparation work is completed, handover must be carried out, which includes two main steps. Firstly, the participating processes respond to the submission preparation sent by the application process. Secondly, send a message to the commitment process indicating that the transaction has been successfully completed, and send appropriate instructions to ensure that all participating processes receive this message, thereby achieving the goal of a formal commitment transaction.

The distributed big data inconsistency detection method points out that existing detection methods can only be applied to centralized data and have low efficiency. In the context of big data, an inconsistency detection algorithm for single or multiple features has been proposed to improve the efficiency of feature based inconsistency detection. Hash function is used to redistribute the data to ensure the accuracy of the detection results and the parallel execution of the algorithm. Since the recognition problem is NP (non deterministic polynominal) - hard, it provides a near optimal solution. For the anomaly detection problem with multiple feature dependencies, clustering and batch parallel detection were conducted based on the structural characteristics of feature dependencies, and optimization problems related to clustering were studied. This paper proposes a general, distributed, parallel and multi-functional dependency conflict detection method based on equivalence class, and proposes a cost model for response time detection and an algorithm for optimizing task allocation. The dynamic load balancing problem is classified as a quadratic programming problem and an approximate optimal solution is found using the Lagrange operator method.

Distributed databases typically use an appropriate level of data redundancy and store copies of data in multiple locations to ensure that the system can continue to operate even if data in one location is compromised, improving system reliability, availability, and performance. It can also reduce communication costs by allowing the system to select and use data replicas that are closest to the user. However, data redundancy can lead to inconsistencies between replicas, and the problem is how to effectively maintain data integrity between multiple replicas.

Definition 1: Inconsistent database. For any integrity constraint $\sum$ on D, the following relationship holds: if $\exists I \square \sum$, then D is an inconsistent database.

Define 2 query integrity constraints. If the entity rule corresponds to the integrity constraint set $\sum$ of pattern R, it cannot be created as $\sum$. If the query Q on R must meet $\sum$, then $\sum$ is called a query integrity constraint.

Definition 3 clustering: Divide relationship R into k tuple subsets Ci($1 \le i \le k$), Ci including multiple tuples with a probability of Pij, tij. If: 1) C1UC2U...UCk=R; 2)$Ci \cap Cl = \theta \wedge i \neq t \wedge l \le i,t \le k$; 3)$tij \in Ci \wedge pi1+pi1+...., pij=1 \wedge 1 \le i \le k \wedge 1 \le j \le |Ci|$; Then Ci is called clustering. The number of tuples within cluster C is called the clustering cardinality, denoted by |Ci|.

Definition 4: A trusted cluster (BC) is cluster C on relationship R, with a cardinality greater than 1, where t is a tuple in C and Q is a query on R. If there is $t \in Q(R)$, cluster C is trusted. If there is any $t \in Q(R)$, cluster C is trusted and is called a trusted cluster. Otherwise, it is called an untrusted cluster.

Define the probability of 5 trusted clusters. Assuming that relationship R is a non-uniform relationship and t is a tuple in trusted cluster C, then the clustering probability of query Q in relationship R is $p=\sum t \in p(t)$. Especially, if the cluster isfully trusted, the probability would be 1.

Mode R is a set of attribute names {A1, A2,.... An}. One of the attributes is a probability attribute, represented by p, corresponding to the numerical range Di of each attribute Aj ($1 \le j \le n$). In the proposed method, there is no unreasonable clustering, that is, the probability attribute value is located in the interval (0,1), and if Aj is p, then Di is located in the interval (0,1).

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

2.2 Consistency Algorithm

Ant colony

At present, heuristic search algorithms based on ant colony, such as particle sieve optimization, ant algorithm and Firefly algorithm, are widely used to determine the diversity of signals [7-8]. Ant algorithm is an idealized ant colony based algorithm, which is similar to that ants determine the shortest path to the ant colony or food source according to the pheromone sent by other ants.

The algorithm can be used to optimize various problems, including motion problems, data classification and text mining: 1) Ants feed on different roads, and each road emits different pheromone. 2) Ants are attracted by the smell of pheromone, so roads with more pheromone are often patronized; 3) Pheromone disappears over time, so roads with less pheromone are often visited. The algorithm calculates the initial pheromone quantity of each element in the pheromone matrix in four steps (Formula (1)).

$$\tau_0 = \frac{fitness(sbest)}{problemsize} \tag{1}$$

A probability function is used to determine the path (formula (2)). P (r, s) represents the probability of reaching ant i from point r. τ (r, s) represents the number of pheromone on the path. Finally, Ji (I) represents the set of training nodes, and l is the last node.

$$P_i(r,s) = \left\{ \frac{[\tau(r,s)]^{\partial} \times [b(r,s)^m]}{\sum_{\mu \partial J_i(I)} [\tau(r,s)]^{\partial} \times [b(r,s)]^m} \right\} \tag{2}$$

Solve the problem of a single exploratory trajectory for ants. The local update of pheromone table can be calculated by formula (3). σ is the local update coefficient, τo is the initial number of pheromone, and τ(r, s) is the value of element (r, s) in the pheromone table.

$$\tau(\mathrm{r},\mathrm{s}) = (1-\sigma) \times \tau(r,s) + \sigma \times \tau_0 \tag{3}$$

Use formula (4) to calculate each element of the total pheromone update table, where p is the total pheromone update rate (0.1 here). Calculate the difference Δτ (r, s) using formula (5).

$$\tau(r,s) - (1-p) \times \tau(r,s) + p \times \Delta\tau(r,s) \tag{4}$$

$$\Delta\tau(r,s) - \{fitness(sbest)\} \tag{5}$$

The steps of genetic algorithm [9-10] are shown in Figure 1.

The specific steps for a distributed database network using genetic algorithms are as follows

(1) Effectively generate the topology of distributed database networks [11-12].

(2) Initialize the relevant parameter values of the genetic algorithm [13-14].

(3) The initial population of chromosomes is generated by randomly searching for the next point from the starting point, and then randomly searching for the next point, just like a walking person [15-16].

(4) Use probability and crossover operations to start iteration, resulting in the latest chromosome variation and creating new individuals during this process [17-18].

(5) Decode chromosomes, convert them into query paths, and calculate the objective function.

(6) Calculation (i.e. communication cost), used to calculate the fitness value for a given chromosome [19].

(7) Using the roulette wheel method, select based on the specific fitness values of each chromosome and proceed to the next iteration.

(8) Determine whether the final conditions are met; If satisfied, generate the optimal trajectory and proceed to the next step; If not satisfied, return to the next step.

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

(9) Initialize the pheromone array using the optimal navigation path [20].

(10) Initialize the parameter values of the polynomial ant colony algorithm.

(11) Genetic algorithm is used to effectively determine the initial pheromone distribution.

(12) Perform these steps separately for different ant colonies.

(13) Set the initial node to be the same as the node in the network that sent the request.

(14) Transfer nodes according to the transmission probability formula while updating the routing.

(15) Check if the ant has completed navigating all nodes at the destination. If so, check if all ants in the ant colony have completed navigation. If not, return to this step.

(16) If all ants in the ant colony have completed the exploration, find the objective function value for a specific path.

(17) Check whether the current number of iterations exceeds the number of iterations that the ant colony runs the pheromone smoothing mechanism. If so, run the pheromone smoothing mechanism.

(18) Determine whether the current iteration count meets the termination condition. If not, return to step.

(19) If not satisfied, provide results.

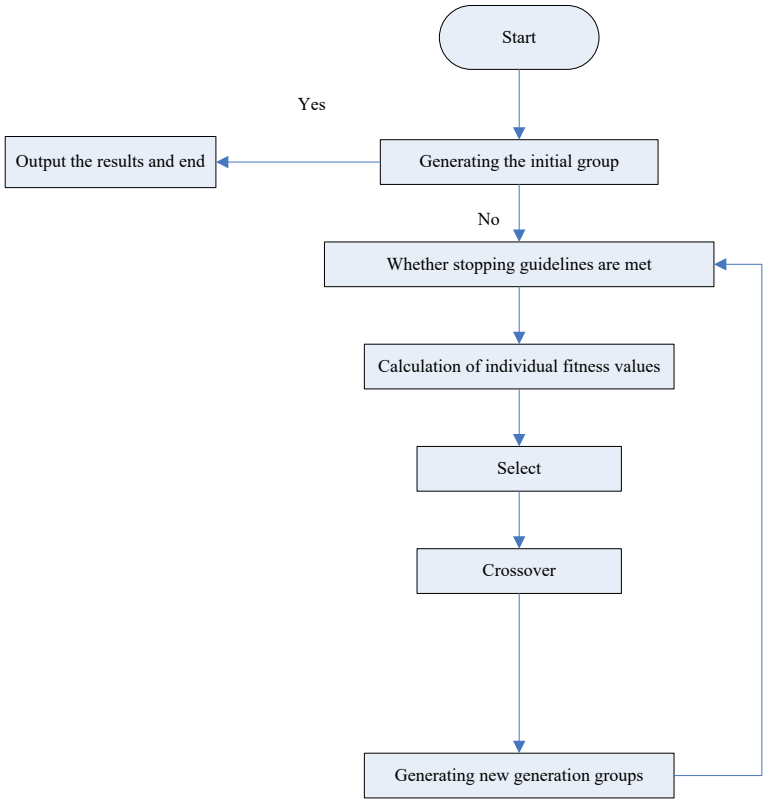

Figure 1: Steps of genetic algorithm.

3. Distributed Database Network Simulation Experiment Based on Genetic Algorithm

In order to test the effectiveness of the algorithm proposed in this article and compare it with the basic genetic algorithm, it used the data environment in reference [9], with the hardware environment being i3 CPU (Central Processing Unit) and the memory being 4GDDR3 (double data rate 3 synchronous dynamic). The hard disk capacity is 1000G, the software is Matlab2010, and the operating system is WindowsXP. This article uses the libraries of our school and surrounding schools to simulate and obtain database query results. In order to achieve database query results, the paper took the libraries of our school and several surrounding schools as the research objects and simulated the data query environment in a distributed database, as shown in Figure 2 [21].

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

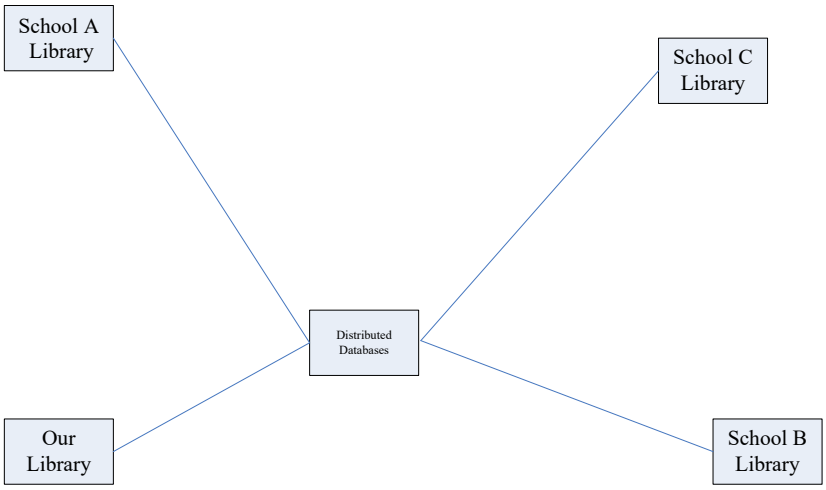

Figure 2: Building a distributed database environment.

From Figure 3, it can be seen that our algorithm has significant advantages in database query efficiency compared to basic genetic algorithms. Our algorithm can improve the query performance of distributed databases.

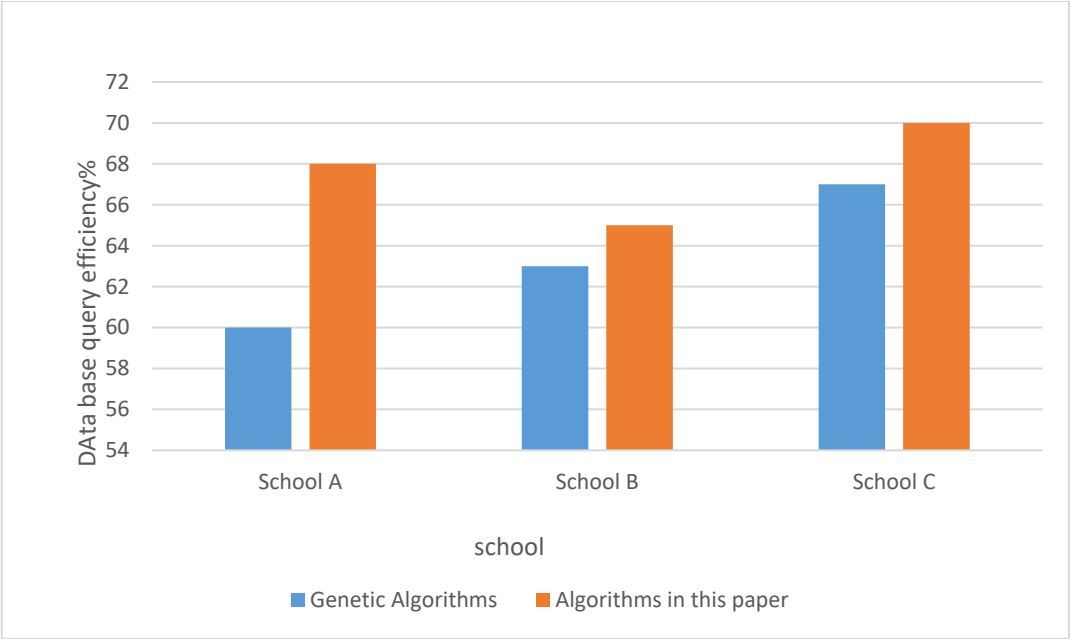

Figure 3: Comparison of two algorithms in database query efficiency.

Figure 4 shows the effect of accessing records from three school libraries in two algorithms. Figure 4 shows the comparison of the access time between the two algorithms with the same number of access records (set to 1000). From Figure 4 above, it can be seen that our algorithm has significant advantages in database access time compared to basic genetic algorithms. Our algorithm can improve the query performance of distributed databases [22].

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

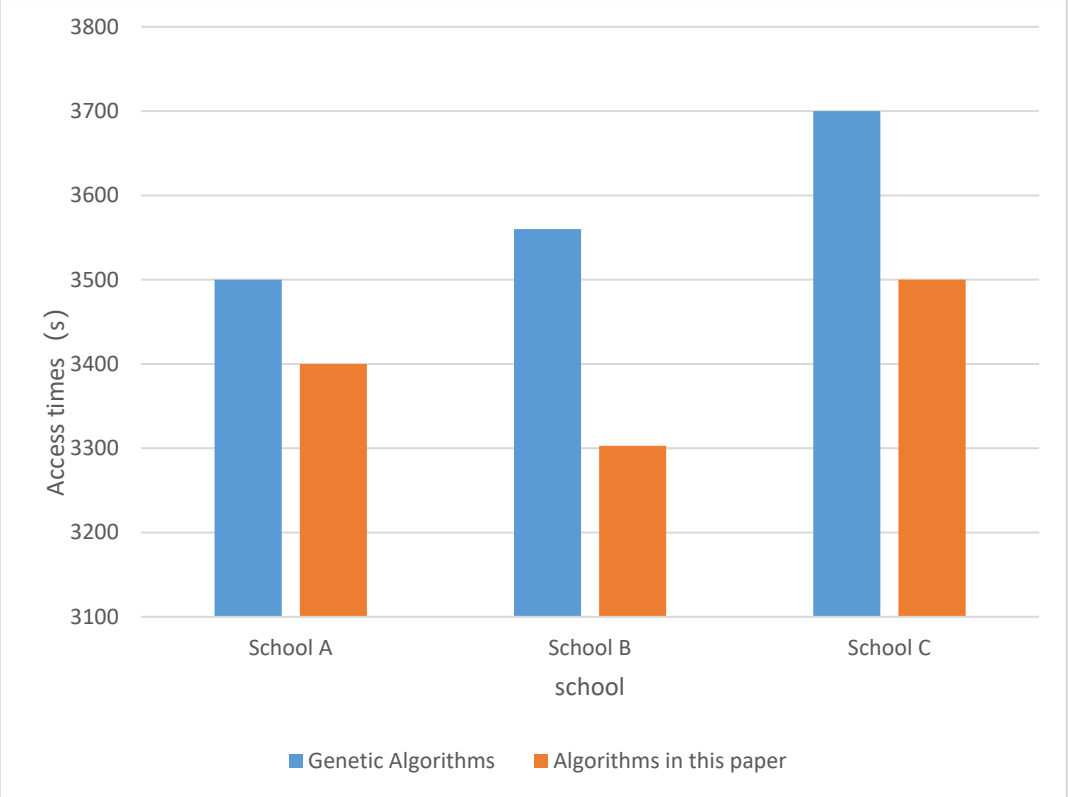

Figure 4: Comparison of access times between two algorithms with the same number of records.

4. Conclusions

The research on consistency processing technology is of great significance in distributed databases. Both methods can fully control the consistency of data. However, by comparing the two strategies, it found that the copy server strategy has significant advantages, that is, it would not affect the normal operation of other websites due to server failure, and it can also ensure data synchronization after recovery. The algorithm discussed in this article has significant advantages over basic genetic algorithms in terms of database query efficiency and database access time.

References

[1] Sattler F, Müller K R, Samek W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE transactions on neural networks and learning systems, 2020, 32(8): 3710-3722.
[2] Lu Y, Huang X, Dai Y, et al. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Transactions on Industrial Informatics, 2019, 16(6): 4177-4186.
[3] Kaur H, Pannu H S, Malhi A K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. ACM Computing Surveys (CSUR), 2019, 52(4): 1-36.
[4] Gu R, Zuo Z, Jiang X, et al. Towards Efficient Large-Scale Interprocedural Program Static Analysis on Distributed Data-Parallel Computation. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(4):867-883.
[5] Abe T, Yamazaki Y, Satoh S. Development of pre-disaster GIS database system for community based urban recovery. AIJ Journal of Technology and Design, 2019, 25(59):377-382.
[6] Guo F, Zhang Q, Zhang Q, et al. Development of a new centralized data acquisition system for seismic exploration. Geoscientific Instrumentation Methods and Data Systems, 2020, 9(1):255-266.
[7] Muthukrishnan S. Influence Sets Based on Reverse Nearest Neighbor Queries. Proc.acm Sigmod Int.conf.on Management of Data, 2019, 29(2):201-212.
[8] Dawid H, Kopel M. on economic applications of the genetic algorithm: a model of the cobweb type. Journal of Evolutionary Economics, 2019, 8(3):297-315.*
[9] Xia G, Chen J, Tang X, et al. Shift quality optimization control of power shift transmission based

ISSN 2616-5775 Vol. 6, Issue 7: 160-167, DOI: 10.25236/AJCIS.2023.060721

on particle swarm optimization–genetic algorithm:. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2022, 236(5):872-892.
[10] Martowibowo S Y, Damanik B K. Optimization of Material Removal Rate and Surface Roughness of AISI 316L under Dry Turning Process using Genetic Algorithm. Manufacturing Technology, 2021, 21(3):373-380.
[11] Chen Y. Location and path optimization of green cold chain logistics based on improved genetic algorithm from the perspective of low carbon and environmental protection. Fresenius Environmental Bulletin, 2021, 30(6):5961-5973.
[12] Vinothini C. Hybrid of Meta Heuristic Firefly and Genetic Algorithm for Optimization Approach in the Cloud Environment. Webology, 2020, 17(1):297-305.
[13] Wu Y M, Li Z, Sun C, et al. Measurement and control of system resilience recovery by path planning based on improved genetic algorithm:. Measurement and Control, 2021, 54(7-8):1157-1173.
[14] Inoue A, Yamamoto N, Nakamura Y, et al. Optimization of Ion Thruster Grids Using JIEDI Code with Genetic Algorithm. Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan, 2021, 19(1):75-80.
[15] Wan M, Zhang S, Song Y, et al. Case Optimization Using Improved Genetic Algorithm for Industrial Fuzzing Test. Intelligent Automation and Soft Computing, 2021, 28(3):857-871.
[16] Zhou J, Ma Q. Establishing a Genetic Algorithm-Back Propagation model to predict the pressure of girdles and to determine the model function:. Textile Research Journal, 2020, 90(21-22):2564-2578.
[17] Knight S J L, Horsley S W, Regan R, et al. Development and clinical application of an innovative fluorescence in situ hybridization technique which detects submicroscopic rearrangements involving telomeres. European Journal of Human Genetics, 2019, 5(1):1-8.
[18] Na A, Fa A, Dsc C, et al. Development and validation of a hybrid aerodynamic design method for curved diffusers using genetic algorithm and ball-spine inverse design method. Alexandria Engineering Journal, 2021, 60(3):3021-3036.
[19] Kakkar M, Bhatti J, Malhotra R, et al. Availability Analysis of an Industrial System under the Provision of Replacement of a Unit using Genetic Algorithm. Gender Technology and Development, 2020, 9(1):1236-1241.
[20] Bhat A T, Anupama, Akshatha, et al. Traffic violation detection in India using genetic algorithm. Global Transitions Proceedings, 2021, 2(2):309-314.
[21] Hirsch Manohar. Design of Distributed Database System Based on Improved DES Algorithm. Distributed Processing System (2022), Vol. 3, Issue 4: 19-27.
[22] Suzanana Ahmad. Implementation Model of Dynamic Load Balancing in Distributed System Based on Consistent Hash Algorithm. Distributed Processing System (2021), Vol. 2, Issue 3: 32-40.