Agricultural & Forestry Economics and Management (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/agrfem.2024.070203
ISSN 2616-2202 Vol. 7 Num. 2

Research on Agricultural Data Processing Based on MySQL

Quanfen Liu[1], Jingjing Wu[2,*]

[1]*School of Computer and Information Engineering, Nantong Institute of Technology, Nantong, Jiangsu, 226000, China*
[2]*Business School, Nantong Institute of Technology, Nantong, Jiangsu, 226000, China*
[*]*Corresponding author*

Keywords: Agricultural Data Processing, MySQL Database, Large-Scale Data, Query Optimization

Abstract: The purpose of this paper is to explore how MySQL database technology can be used to process and analyze agricultural data in order to improve data management efficiency and crop yield. By analyzing existing studies and methods, this paper proposes a novel data processing scheme based on MySQL and verifies its effectiveness through experiments. In the processing speed evaluation experiment, when the word data size increased from 100000 to 5 million, the import time based on MySQL database increased from 12 seconds to 580 seconds, and the query time increased from 1.2 seconds to 72 seconds. And the query accuracy always remains above 99%. The experimental results show that this MySQL -based method can effectively improve the speed and accuracy of processing large-scale agricultural data. In addition, we also hope to apply this method to more types of agricultural data processing scenarios to test its adaptability and practicality in different situations.

1. Introduction

In this paper, we explore effective methods for processing agricultural data using MySQL database technology, aiming to improve data management efficiency and optimize crop yield. With the widespread application of digital technology in agriculture, how to effectively manage and analyze a large amount of agricultural data has become a challenge. By utilizing MySQL, we can better organize, query, and analyze this data, which is of great significance for improving the scientific and precise nature of agricultural production.

This paper mainly proposes a data processing scheme based on MySQL and verifies its effectiveness through a series of experiments. We have explained in detail every step from data collection and preprocessing to database design, data import, and query optimization. Especially in terms of database design and optimization queries, we have adopted some efficient technical strategies, which have significantly improved the speed and accuracy of data processing, demonstrating the practicality and efficiency of this method in modern agricultural data processing.

The structure of the paper is as follows: Firstly, the introduction section introduces the research background and purpose. Next is a review of relevant work, comparing and analyzing the current

research status and existing problems. The third part provides a detailed introduction to our method, including data collection, preprocessing, database design, and optimization. The fourth part is experimental design and result analysis, which verifies the effectiveness of the method. The final conclusion summarizes the research findings and proposes further research directions. This structural arrangement helps readers to systematically understand the research content and experimental process.

2. Related Work

Currently, many researchers are devoted to the study of agricultural data processing. For example, Zhao Chunjiang analyzed the agricultural knowledge intelligent service technology such as perception recognition, knowledge coupling, and reasoning decision-making, and constructed an agricultural knowledge intelligent service platform composed of cloud computing, big data processing framework, knowledge organization management tools and knowledge service application scenarios [1]. Big data in agriculture is a set of techniques to address the challenges of the new data era. Cravero A developed a framework that summarized the main challenges, machine learning techniques used and leading technologies [2]. In identifying the evolutionary patterns of big data and AI methods in wind energy forecasting, Zhao E summarized the research on big data and AI methods in wind energy forecasting over the past two decades [3]. Digitization has impacted agriculture and food production systems, enabling the application of technology and advanced data processing techniques in agriculture. Nasirahmadi A described the recent advances in the concept of digital twins, and different digital technologies and techniques in the agricultural context [4]. However, their approach suffers from performance bottlenecks when dealing with large-scale data. The review by Debauche O provided a panoramic comparison of centralized clouds, distributed cloud architectures, collaborative computing strategies and new trends in agriculture 4.0 [5]. The convergence of IoT and cloud technologies has driven the development of smart agriculture and revolutionized modern agricultural practices. Khan A proposed a step-by-step framework for optimizing onion crop management using IoT sensors and cloud solutions [6]. In recent years, governments have increasingly focused on online learning platforms. Chang J H built a high-performance digital learning platform for agriculture aiming to achieve learning diversity, improve users' ability and willingness to learn, and break through geographic constraints [7]. Their research has achieved some results, but their methodology relies on costly computational resources. Overall, the existing studies are still deficient in processing efficiency and cost control.

In order to improve the efficiency and reduce the cost of agricultural data processing, many researchers have tried various approaches. For example, the Internet of Things (IoT) technology has revolutionized almost every industry, including “smart farming” or “precision farming”. Akhter R proposed a model for predicting apple diseases in apple orchards of Kashmir valley using data analytics and machine learning in IoT systems [8]. Research on sustainable computing in agriculture has great potential as an effective way to address most of the technological bottlenecks in agriculture. Nie J systematically presented the classification and application of relevant algorithms in the field of sustainable computing in agriculture [9]. However, their methods have challenges in data consistency management. Other researchers such as Fei S utilized a machine learning approach to fuse UAV-based multi-sensor data to improve the accuracy of crop yield prediction [10]. However, the complexity of their algorithms is high and difficult to be applied in resource-limited environments. In this paper, we propose a MySQL-based agricultural data processing method to address the performance and cost deficiencies of existing methods.

3. Methods

3.1 Data Acquisition and Preprocessing

In the data acquisition and pre-processing section, we collect the data generated during the agricultural production process in real time through sensor networks and remote monitoring systems. These data include soil moisture, meteorological data, crop growth records and many other types. In order to ensure the quality and consistency of the data, we first pre-process the collected raw data. The preprocessing process includes data cleaning, data filtering and data normalization.

One of the main purposes of data cleaning is to remove noise and outliers from the data. Some simple but effective rules are used to identify and remove these outliers, for example, values read by the soil moisture sensor in a specific time period that are outside of a reasonable range are marked as outliers and removed. Data filtering, on the other hand, is designed to remove redundant and duplicate data. During data collection, data may be duplicated due to network transmission or sensor failure. We identify duplicate records and remove them by comparing timestamps and data values.

Standardization is one of the important steps in data preprocessing. Different sensors may use different measurement units and accuracies, and in order to unify the processing in the subsequent database operation and data analysis, we converted all the data into units and standardized the accuracies. For example, temperature data were converted to degrees Celsius uniformly, and soil moisture data were converted to percentage representation. Meanwhile, in order to facilitate the subsequent data analysis, we unified the formatting of time stamps to ensure that all data records have precise time stamps.

After preprocessing, the data is imported into the MySQL database. During the data import process, we designed an efficient batch import method to improve the efficiency of data import. Table 1 shows some of the preprocessed soil moisture data:

Table 1: Soil moisture data

Timestamp	Sensor ID	Soil Moisture (%)
2024/5/1 8:00	S1	23.5
2024/5/2 8:00	S2	30.2
2024/5/3 8:00	S1	22.8
2024/5/4 8:00	S2	29.7
……	……	……

The data in Table 1 go through these steps to ensure data accuracy and consistency, providing a reliable data base for subsequent analysis and decision-making.

3.2 Database Design

The efficient storage and fast querying of agricultural data are key points in the database design process. The system needs to handle large-scale data while maintaining performance and scalability. Therefore, we have designed the database structure. The use of relational database management system MySQL is due to its stability, performance advantages, and functional support when handling large-scale data. Database pattern design is the core, which divides the database into several main tables based on the characteristics of agricultural data, including soil moisture data table, meteorological data table, and crop growth record table. Each table contains detailed field design to ensure data integrity and accuracy. Soil moisture data table includes fields such as timestamp, sensor ID, and humidity value. These fields not only record the data itself, but also provide necessary metadata support for subsequent data analysis and processing. The specific

situation is shown in Table 2:

Table 2: Data model of soil moisture meter

Field Name	Data Type	Description
Timestamp	DATETIME	Record time
Sensor ID	VARCHAR	Identifier of sensor
Soil Moisture	FLOAT	Soil moisture value

In order to improve the query efficiency, we introduced indexes in the data schema for designing the soil moisture table in Table 2. The query efficiency can be expressed by equation (1):

$$\mathrm{QE} = \mathrm{O}(\log(\mathrm{n})) \tag{1}$$

Where in equation (1), n denotes the size of the dataset, this equation shows that the query cost of using B-tree indexes is on logarithmic level, which significantly improves the query performance.

The use of indexes can significantly reduce query times, especially when working with large-scale data. For example, we created composite indexes on the timestamp and sensor ID fields, which allowed the system to quickly locate data for specific time periods and specific sensors. In addition, we created separate indexes for commonly used query fields to further optimize query performance.

In addition to index design, we normalized the data tables. Through normalization, we eliminated data redundancy and ensured data consistency and integrity. For example, the temperature, humidity and wind speed fields in the weather data table are split into separate sub-tables to reduce duplicate storage and improve the efficiency of data update. Meanwhile, we utilize foreign key constraints to maintain the association relationship between different tables to ensure the referential integrity of the data.

In terms of data storage, we considered the special characteristics and storage needs of agricultural data. Since agricultural data is time-continuous and real-time, we chose the partitioned table technique to store the data in partitions by time periods. This approach not only helps to improve query performance, but also facilitates archiving and management of historical data. For example, storing a year's worth of data partitioned by month allows the system to query a particular month's data by accessing only the corresponding partition, thus greatly reducing query time [11].

3.3 Data Import

In the data import session, we first chose the bulk data import tool provided by MySQL, such as the LOAD DATA INFILE command. This command can quickly import a large amount of data from a text file much faster than a single line insertion. In order to improve the efficiency of data import, we will be the original data preprocessing, first converted to CSV format files, and then imported using the LOAD DATA INFILE command. This method is not only easy to operate, but also shows high efficiency in practical application.

In the process of data import, we used the batch import technology, which can be expressed by equation (2):

$$T_{batch} = \frac{N}{B} \times T_{single} \tag{2}$$

In equation (2), T_{batch} represents the batch import time, N represents the total number of records, B is the batch size, and T_{single} represents the single record import time.

We also paid special attention to data consistency and integrity. To prevent errors or interruptions in the data import process, we performed strict format checks on the data files before import to

ensure that the format and content of each row of data conformed to the requirements of the database tables. In addition, to ensure data consistency, we use transaction processing when importing data. By placing the data import operation in a transaction, if an error occurs during the import process, all operations can be rolled back, avoiding the problem of data inconsistency caused by partial data import.

Different types of data have been designed with different import strategies. To improve data import performance, we disabled some indexes and constraints during the import process. Although indexes and constraints are important in data queries, they may significantly reduce speed during data import. We temporarily disable these indexes and constraints before importing, and re enable and rebuild the indexes after data import is completed, which greatly improves the speed of data import while ensuring data consistency and integrity [12].

3.4 Query Optimization

We have created indexes for common query fields when designing the database. For example, for key fields such as timestamp, sensor ID, and crop ID, we created B-tree indexes. These indexes significantly speed up data retrieval, allowing query operations to be completed in less time. In particular, the indexes on the timestamp field provide a particularly significant performance improvement for time-range queries. In addition, in order to avoid full table scans, we also created some composite indexes that are optimized for multi-field combination queries.

We also utilize MySQL's query caching feature. In some frequently repeated queries, query caching can significantly reduce the query time. By caching the results of recently executed queries, when the same query is executed again, the results can be read directly from the cache without re-accessing the database tables. To ensure effective cache utilization, we regularly monitor the hit rate of the query cache and adjust the cache size and expiration policy according to the actual situation to maximize the hit rate of the cache.

In the actual query optimization process, we also rewrote and optimized some complex queries. For example, for some join queries involving multiple tables, we identified and optimized potential performance bottlenecks by analyzing the execution plan. We used the EXPLAIN statement provided by MySQL to view the execution plan of the query to determine the execution order and index usage of each operation step in the query. By analyzing the execution plan, we identified inefficient parts of the query and optimized them, such as adjusting the table join order, using more appropriate indexes, and even breaking some complex queries into multiple simple queries for execution.

For query optimization under large data volume, we also introduce partitioned table technology. Partitioning a large table by time or other dimensions makes the query only need to access the relevant partition, which reduces the amount of data scanning and significantly improves the query performance. Taking soil moisture data as an example, we partition it by month, so that when querying the data of a certain month, only the corresponding partition needs to be scanned, which avoids the performance problems caused by full table scanning [13-14].

4. Results and Discussion

4.1 Processing Speed Evaluation Experiment

The purpose of this experiment is to evaluate the performance of MySQL-based agricultural data processing methods. Different sizes of agricultural data sets (100,000, 500,000, 1,000,000, 5,000,000, and 10,000,000 records) are generated by simulation, and the import time and query time of each data set are measured. The data types include soil moisture, meteorological data and crop

growth records. This is shown in Figure 1:

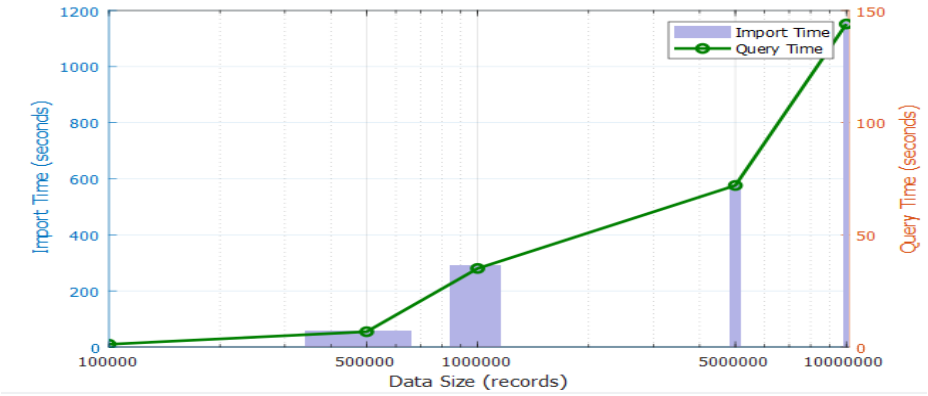

Figure 1: Evaluation of processing speed

As shown in Figure 1, at a data size of 100,000 records, the data import time is 12 seconds and the query time is 1.2 seconds. When the data size reaches 5 million records, the data import time is 580 seconds and the query time is 72 seconds. From the conclusion of the data, it can be seen that the performance of MySQL-based large-scale agricultural data processing is superior, and it can effectively improve the efficiency of data processing to meet the needs of modern agricultural production.

4.2 Data Accuracy Assessment Experiment

In the data accuracy evaluation experiment, we evaluated the accuracy of the MySQL based agricultural data processing method. We will compare the query results with the expected results after the experiment and calculate the accuracy. We used MATLAB to plot the variation of accuracy with data size.

In Figure 2, at a data size of 100,000 records, the query accuracy is 99.8%. When the data size reaches 10 million records, the query accuracy is 99.1%. From the data results, it can be seen that the accuracy rate decreases slightly with the increase of data size, but the overall rate remains above 99%, which proves the reliability and accuracy of MySQL in large-scale agricultural data processing. The specific data is shown in Figure 2:

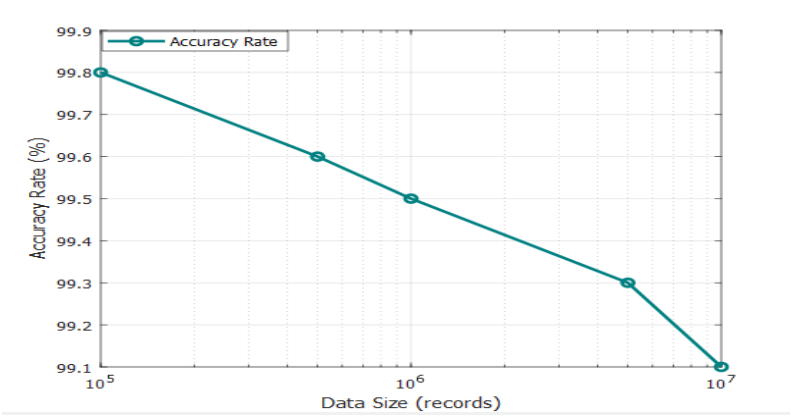

Figure 2: Data accuracy evaluation

4.3 Resource Use Evaluation Experiment

The resource utilization evaluation experiment aims to evaluate the system resource utilization of

MySQL based agricultural data processing methods at different data scales. In the experiment, we measured the CPU usage and memory usage during import and query operations. The specific data situation is shown in Figure 3:

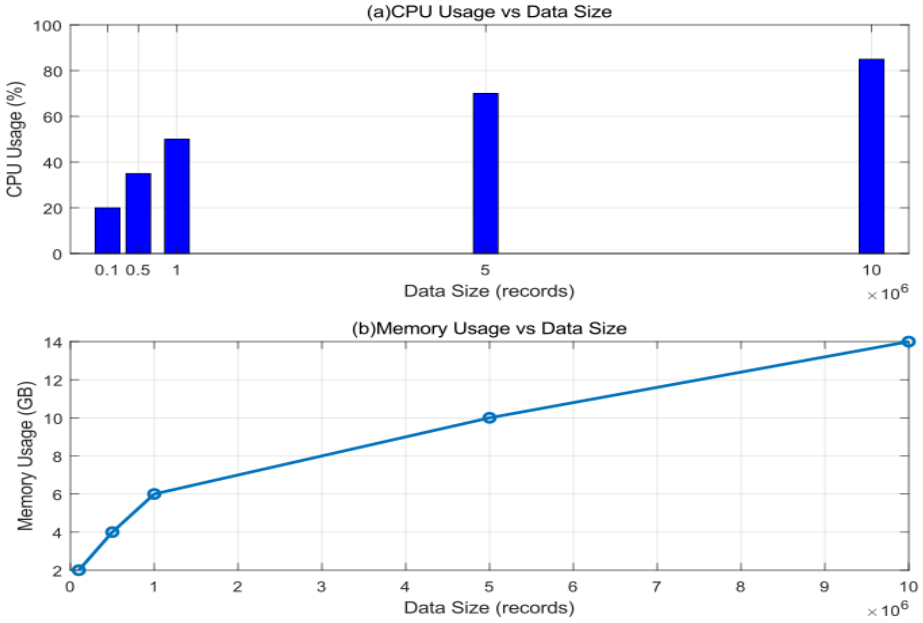

Figure 3: Resource use assessment

Figure 3(a) represents the CPU utilization and Figure 3(b) shows the memory usage. In Figure 3, at a data size of 100,000 records, the CPU utilization is 20% and the memory usage is 2 GB. At a data size of 10 million records, the CPU utilization is 85% and the memory usage is 14 GB. The conclusion from the data shows that with the increase of data size, the CPU and memory utilization shows a linear growth trend, which suggests that MySQL has a high demand on system resources when processing large-scale agricultural data with high demand on system resources.

5. Conclusion

This paper introduces a method of using MySQL to process agricultural data. We have not only demonstrated the effectiveness of this method through theory, but also through practical experiments. In the paper, we elaborated on the various steps from data collection and preprocessing to database design, data import, and query optimization. The experimental results show that this MySQL based method can effectively improve the speed and accuracy of processing large-scale agricultural data. Although our method has shown good performance in current experiments, we also realize that the demand for system resources significantly increases when dealing with extremely large amounts of data. This may limit the application of this method in situations where resources are limited. Therefore, future work will focus on how to further optimize resource utilization and improve processing efficiency. In addition, we also hope to apply this method to a wider range of agricultural data processing scenarios to test its adaptability and practicality in different situations.

Acknowledgement

This work was supported by Nantong University of Technology College Student Innovation and Entrepreneurship Training Project, Project Number (XDC2023123), Project Name: Agricultural

Planting Data Processing - π Agriculture; Young and middle-aged backbone teachers at Nantong University of Technology, project number: ZQNGGJS202219.

References

[1] Zhao Chunjiang. A review of agricultural knowledge intelligent service technologies[J]. Intelligent Agriculture in English, 2023, 5(2):126-148.
[2] Cravero A, Pardo S, Sepúlveda S, et al. Challenges to use machine learning in agricultural big data: a systematic literature review[J]. Agronomy, 2022, 12(3): 748-761.
[3] Zhao E, Sun S, Wang S. New developments in wind energy forecasting with artificial intelligence and big data: A scientometric insight[J]. Data Science and Management, 2022, 5(2): 84-95.
[4] Nasirahmadi A, Hensel O. Toward the next generation of digitalization in agriculture based on digital twin paradigm[J]. Sensors, 2022, 22(2): 498-512.
[5] Debauche O, Mahmoudi S, Manneback P, et al. Cloud and distributed architectures for data management in agriculture 4.0: Review and future trends[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(9): 7494-7514.
[6] Khan A, Hassan M, Shahriyar A K. Optimizing onion crop management: A smart agriculture framework with iot sensors and cloud technology[J]. Applied Research in Artificial Intelligence and Cloud Computing, 2023, 6(1): 49-67.
[7] Chang J H, Chiu P S, Lai C F. Implementation and evaluation of cloud-based e-learning in agricultural course[J]. Interactive Learning Environments, 2023, 31(2): 908-923.
[8] Akhter R, Sofi S A. Precision agriculture using IoT data analytics and machine learning[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(8): 5602-5618.
[9] Nie J, Wang Y, Li Y, et al. Sustainable computing in smart agriculture: survey and challenges[J]. Turkish Journal of Agriculture and Forestry, 2022, 46(4): 550-566.
[10] Fei S, Hassan M A, Xiao Y, et al. UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat[J]. Precision agriculture, 2023, 24(1): 187-212.
[11] Upadhyaya A, Singh A K, Kumar S, et al. Development of relational data base management information system on integrated farming: Data base information system for integrated farming[J]. Journal of AgriSearch, 2022, 9(4): 342-346.
[12] Ngo V M, Duong T V T, Nguyen T B T, et al. A big data smart agricultural system: recommending optimum fertilisers for crops[J]. International Journal of Information Technology, 2023, 15(1): 249-265.
[13] Anass D, Madi A A, Alihamidi I, et al. A novel autonomous remote system applied in agriculture using transmission control protocol[J]. Int J Reconfigurable & Embedded Syst, 2022, 11(1): 1-12.
[14] Wu J, Pichler D, Marley D, Wilson D, Hovakimyan N, & Hobbs J. Extended Agriculture-Vision: An Extension of a Large Aerial Image Dataset for Agricultural Pattern Analysis. arXiv preprint arXiv:2023, 2303:2460.