Hindawi
Journal of Sensors
Volume 2022, Article ID 4827021, 9 pages
https://doi.org/10.1155/2022/4827021

Research Article

Design of Distributed Human Resource Management System of Spark Framework Based on Fuzzy Clustering

Qing Sun,[1] **Tao Wu**,[1] **and Jia Hua**[2]

[1]*International Cooperation Department, Xuzhou University of Technology, Jiangsu Xuzhou 221018, China*
[2]*Human Resource Management Department, Xuzhou College of Industrial Technology, Jiangsu Xuzhou 221140, China*

Correspondence should be addressed to Qing Sun; sunq@xzit.edu.cn

Received 5 January 2022; Accepted 26 January 2022; Published 11 March 2022

Academic Editor: Yanqiong Li

The construction of human resource management system is a key part of enterprise management and control. A perfect human resource management system is conducive to the long-term development of enterprises. Aiming at improving the current situation of enterprise human resource management, a distributed human resource management system is proposed in this paper based on Spark framework. Aiming at the disadvantages of traditional *k*-means algorithm in processing massive data, such as low computational efficiency and high time complexity, an improved *k*-means algorithm based on Spark computing framework is proposed. Through the spatial location relationship with the cluster center, redundant calculation is reduced and the ability of processing massive data is improved for the system. Combined with the actual situation of the enterprise, the human resource management system architecture is designed by using Java EE human-computer interaction. The proposed system can achieve user management, employee information, attendance, evaluation, performance, salary, personnel change, and other business management. The experimental results demonstrate that the system can effectively reduce the time complexity of calculation and improve the system efficiency.

1. Introduction

With the continuous improvement of the level of social and economic development, many enterprises also began to expand their own team to achieve good development. At present, the rapid development of cloud computing and big data Internet technology also makes people gradually enter the intelligent information era. Many enterprises develop distributed systems to be applied to all kinds of business, such as collaborative office [1], document management [2], financial management [3], and other systems [4]. For the development of enterprises, talent is an important cornerstone for the development of enterprises. It is of great significance to develop and design a set of human resource management system to improve the working efficiency of talents in enterprises through reasonable human resource management [5]. In order to get rid of the obstacles caused by traditional human resource management to the development and expansion of enterprises and provide efficient enterprise human resource development platform, it is necessary to design a perfect human resource management system [6]. In short, the construction of human resource management system is a key component for the development of enterprise management and control. It needs to be fully combined with other construction systems, and the starting point is based on the business form and long-term development strategy of the company [7]. At the same time, when designing the human resource management system, it is necessary to consider the full use of the existing human resources of enterprises. By analyzing the performance requirements of human resource management system design, the system needs to consider business process, data process, data dictionary, use case constraints, etc.

k-means algorithm is an unsupervised learning algorithm and has become one of the most widely used clustering algorithms. With the rapid development of open-source distributed computing framework, clustering algorithm based on distributed computing platform can effectively solve the problem of memory overflow in single-machine mode [8, 9]. This direction has become a research

hotspot. At present, for the problem of parallelization of algorithms under large data sets, many scholars have optimized and realized the algorithm under the distributed framework of MapReduce [10, 11]. Moreover, the optimization method research under Spark framework is relatively few.

Aiming at the problem of high time complexity of k-means algorithm [12], literature [13] improved by introducing Canopy algorithm and maximum and minimum distance method on Spark platform. The convergence speed of the algorithm has been improved, but the problem of large amount of redundant computation has not been solved fundamentally. Literature [14] proposed a clustering algorithm based on distance triangle inequality in cloud computing framework. But the optimization strategy using triangle inequality principle needs to save the upper and lower bound information of each data. It is difficult to fully implement this in the Spark framework. Literature [15] compares the operating efficiency of k-means algorithm based on MapReduce and Spark. Its experimental results show that the Spark framework has a more efficient running speed for the algorithm that needs repeated iteration.

The innovations and contributions of this paper are listed below.

(1) In order to solve the problem of large amount of redundant calculation of k-means algorithm, this paper introduces the spatial location relationship between grid cells and clustering centers

(2) Because the triangle inequality optimization strategy is considered in this paper, the redundant distance calculation is greatly reduced

(3) The improved algorithm is implemented in parallel under the Spark framework. It improves the processing power of large data sets. Finally, the effectiveness of the proposed algorithm is verified by experimental analysis

The structure of this paper is listed as follows. Distributed k-means optimization algorithm based on spark framework is described in the next section. The proposed system is expressed in Section 3. Section 4 focuses on the experiment and analysis. Section 5 is the conclusion.

2. Distributed k-Means Optimization Algorithms Based on Spark Framework

2.1. Spark Distributed Framework. Spark is a commonly used distributed computing platform that can effectively process massive data analysis. It is a distributed computing framework based on Elastic Distributed Data Set (RDD) implementation initiated by AMPLab of UC Berkeley [16].

In the overall structure of Spark, each Spark application uses a driver program to initiate parallel operations on the cluster. The driver program can manage multiple actuator nodes simultaneously. In a distributed cluster environment, multiple working nodes can read data from the HDFS file system and convert it to RDD. An RDD is an immutable, distributed collection of objects. Each RDD is divided into multiple partitions. These partitions run on different actuator nodes.

Compared with the disk-based MapReduce calculation mode, Spark does not need to save the intermediate results of iteration to disks. Thereupon, it has more efficient computing efficiency. The Spark-based algorithm has good scalability and can better adapt to large-scale data sets. In the clustering algorithm which needs many iterations, its advantage is more obvious.

2.2. k-Means Algorithm. The process of the traditional k-means algorithm [17] is as follows.

Input: number of clusters z, data set d.

Output: z class clusters.

The algorithm steps are as follows.

(1) Select z points from data set d as the initial clustering centers

(2) Allocate each data point in d to the nearest class cluster

(3) Calculate the average value of vector coordinates of data points in each class cluster, and then, update the cluster center of this class cluster

(4) Repeat steps (2) and (3) until the clustering center converges

(5) The sum of squares of errors is used as a measure of clustering quality, and it is calculated by the following equation

$$S = \sum_{x=1}^{z} \sum_{d \in D_x} d - c_x{}^2, \tag{1}$$

where z is the number of class clusters and c_x is the cluster center of the x-th class cluster D_x.

The time complexity of k-means algorithm is $\phi(tzw)$, where t is the number of data points, z is the number of class clusters, and w is the number of iterations. The time complexity of the algorithm is relatively large, and affected by the number of class clusters, it increases with the increase of z value.

2.3. Use k-Means Optimized by Spatial Information. In every iterative calculation process of k-means algorithm, each data point needs to calculate the distance between it and k cluster centers. The redundancy of its calculation is great, especially when the value of k is large, which has a great influence on the time efficiency of the algorithm. To solve the problem of large redundancy of k-means algorithm, the more effective improvement strategy is triangle inequality method. Without changing the clustering results of k-means algorithm, it can greatly reduce the computational complexity. The redundancy of calculation can be greatly reduced by using the principle of triangle inequality. However, in a single iteration, for each data point, several distance calculations are still needed to find the nearest class cluster. In

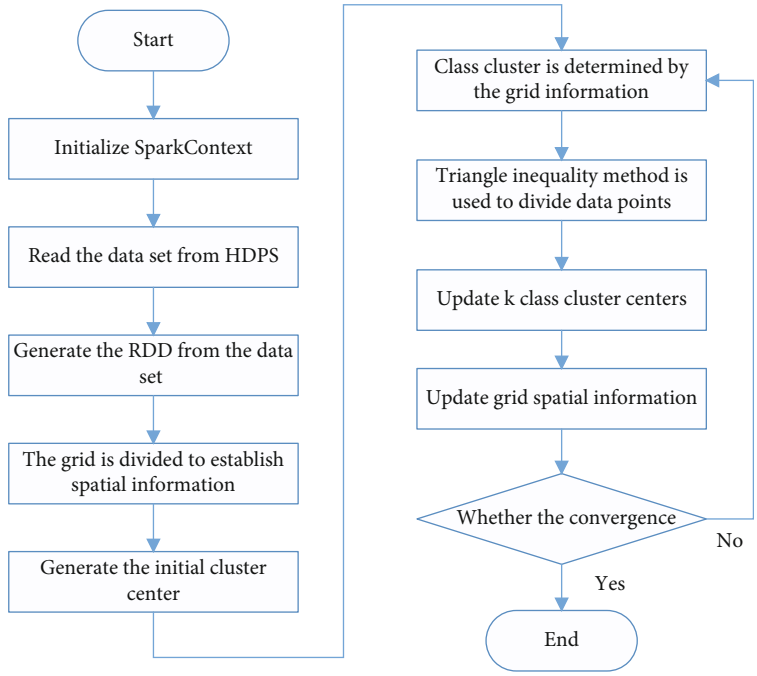

Figure 1: k-means optimization algorithm based on Spark framework.

fact, with the increase of k value, the number of calculations for a single data point will gradually increase, and the redundancy of calculation is positively correlated with k value. Aiming at the shortcomings of the triangle inequality optimization strategy, the spatial position information of data points is further considered to reduce the redundant computation.

2.3.1. Spatial Position Relationship. In order to further reduce the time complexity based on applying the triangle inequality strategy, the algorithm in this paper introduces the spatial position relationship between data points and clustering centers in k-means clustering. The basic idea can be described as follows. For any data point, if the spatial position relationship between it and k cluster centers can be known, then the closest cluster center to it can be accurately determined. Instead of doing k calculations, you just need to assign data points to the corresponding classes. Therefore, it is necessary to design a method that can efficiently save the spatial position relationship between all data points and k cluster centers. In view of the high efficiency of grid partitioning process, it is appropriate to use grid cells to store spatial location information of data points.

2.3.2. Establish Spatial Location Information of Grid and Class Cluster. First, the data set is meshed with a certain partition width in each dimension. Then, the position relationship between each grid containing data points and k cluster centers is judged.

Taking two-dimensional data as an example, for any grid C, the method to determine the spatial position relationship between grid C and k clustering centers is as follows.

(1) First, find the class cluster closest to the center of grid C from z clustering centers and set it as A, whose distance is d_1. Let the concentric circle radius of the grid be r. Therefore, the maximum value of distance A for any point in grid C is $d_1 + r$

(2) Then, calculate the distance of other $k-1$ clustering centers in turn. Taking B as an example, let the distance between B and C be d_2. The minimum value of the distance B of any point in grid C is $d_2 - r$, if the following equation is satisfied

$$d_1 + r < d_2 - r. \tag{2}$$

That is, the closest distance between any point in grid C and B is still greater than the furthest distance between any point in grid C and A, so any point in grid C cannot belong to class cluster B. If the above equation is not satisfied, the score points inside grid C may belong to B. Grid C needs to record all possible belonging class clusters. In fact, when the number of meshes is large enough, the vast majority of meshes will have only one belonging class cluster. Only a small number of grids will belong to more than two class clusters. The average number of belonging class clusters per grid is slightly more than 1. By establishing the spatial location relationship between each grid and k cluster centers, the location relationship between all data points and k cluster centers is obtained.

2.3.3. Clustering Using Spatial Location Information. Take two-dimensional data as an example. Any data point is $w(i, j)$. Let the maximum value of dimension i be $\max_i$ and the minimum value be $\min_i$. When meshing, the number of segments in one dimension is iNum. Let the dimension in which j resides be $\max_j$ at its maximum and miny at its

minimum. In grid division, the number of segments in one dimension is jNum. According to the coordinates of data points, the grid position (i', j') of w can be quickly obtained by following the equation.

$$i' = \left(i - \min_i\right) \div \left(\max_i - \min_i + \delta\right) \times i\text{Num}, \quad (3)$$

$$j' = \left(j - \min_j\right) \div \left(\max_j - \min_j + \delta\right) \times j\text{Num}, \quad (4)$$

where δ is a positive number less than 1.

The relationship between w and class cluster can be determined according to the relationship between the grid where data point w resides and k cluster centers. Namely, what cluster centers w may belong to. In this way, distance calculation between w and all cluster centers can be avoided.

2.3.4. k-Means Optimization Algorithm in Spark Framework. The algorithm flow of this paper is shown in Figure 1. The algorithm is implemented in parallel under the Spark framework. First, initialize Spark Context to determine the data set, number of cluster centers k, and maximum number of iterations. The data set is then read from the HDFS (Hadoop Distributed File System) file system and converted into an RDD collection. The Spark cluster partitions RDD sets based on the Spark Context information. It makes each partition run on each actuator node for grid partitioning and establishes the spatial information of grid and k cluster centers. Select the initial cluster center and start iterative calculation.

In each iteration calculation process, mapPartitions are processed for each RDD partition first. Cluster allocation is made to each data point. The corresponding grid is obtained by data point coordinates. Then, the relationship between grid and class cluster is utilized to determine the $t(t \leqslant k)$ class clusters that data points may belong to. The triangle inequality method is utilized to find the nearest class from t cluster centers (instead of k) to reduce the redundant distance calculation. After all data points were allocated, the class cluster centers of different actuator nodes were summarized by reduceByKey operations. New k clustering centers were obtained, and the spatial relationship between grid and class cluster was updated. Finally, the sum of squares of errors is calculated, and the next iteration is judged.

The key point of the optimization strategy is to use the grid structure to preserve the spatial relationship between all data points and k cluster centers and obtain the possible belonging class cluster of any data points according to this relationship. The effect of the proposed algorithm is similar to that of the triangle inequality, which does not change the final cluster center of k-means algorithm after clustering. This strategy can further reduce the amount of computation based on the application of triangle inequality.

2.3.5. Time Complexity Analysis of the Algorithm. The time complexity of the improved algorithm is effectively reduced. Reasonable selection of grid partition width can ensure that most grids have only one possible cluster. Only a small number of grids may belong to two or more class clusters. In this way, for most data points, only one distance calculation is required, instead of k times. The time complexity of the improved algorithm is $\phi(tw)$, where t is the number of data points and w is the number of iterations.

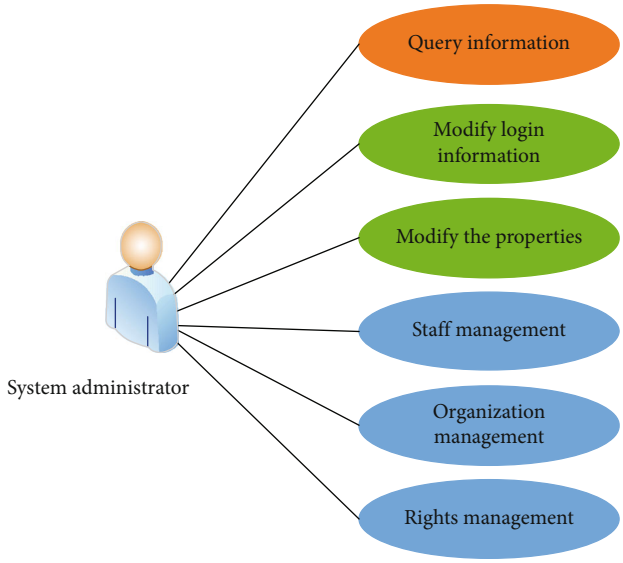

Figure 2: The system case for system administrator.

Compared with the improved strategy of triangle inequality, the combination method using spatial position information has lower time complexity. Especially with the increase of k value, its advantage becomes more obvious.

Therefore, it can avoid a lot of redundant distance calculation process. For any data point, the advantage of using grid-based spatial location information is that most of the distance can be too far. It filters out the cluster center which obviously does not have the belonging relation, avoiding a lot of redundant distance calculation process.

3. The Design of Human Resource Management System

3.1. Analysis of System Design Requirements

3.1.1. Feasibility Analysis

(1) Analyze the system from the perspective of technical feasibility. In order to be able to fully enhance the system application management decision-making level, many large and medium-sized enterprises are vigorously developing human resource management system. However, with the development and expansion of the enterprise team, the existing human resource management system cannot meet the needs of the enterprise. Enterprises began to increase the research and development of human resource management system, the formation of more and more mature human resource management technology. Therefore, Java EE-based human resource management system has technical feasibility

(2) Analyze the system from the perspective of operational feasibility. Design based on Java EE technology human resource management system, which

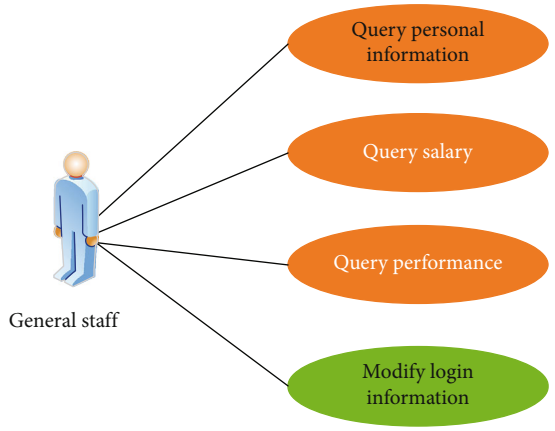

Figure 3: The system case for general staff.

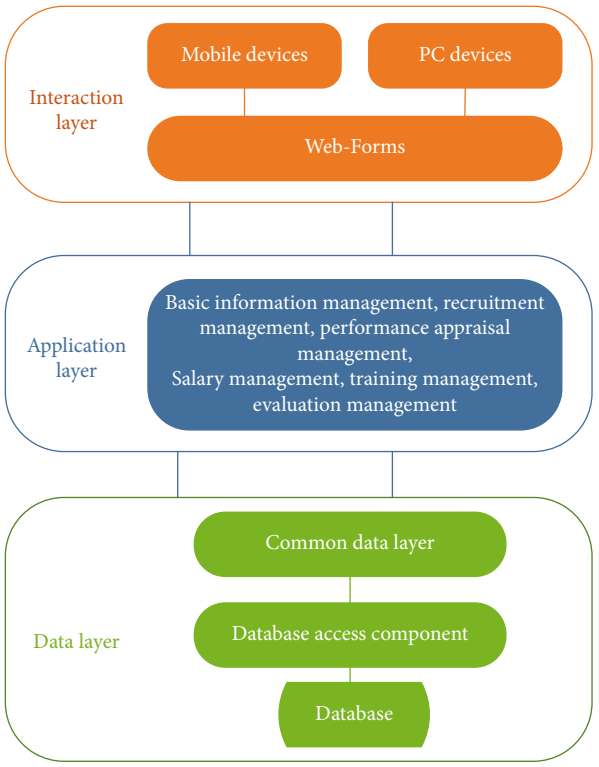

Figure 4: The human resource system business operation architecture diagram.

can facilitate the use of every common employee system. It realizes good man-machine interaction and ensures the feasibility of system development and operation

(3) Analyze the system from the perspective of economic feasibility. The fundamental pursuit of enterprises is social and economic benefits, so how to maximize the benefits of enterprises is very important. And the enterprise's ability to bear the new technology also determines whether the enterprise can ensure the maximum benefit. The design of human resource management system based on Java EE technology can effectively simplify the workflow of human resource management. It can make scientific and reasonable decision in real time and ensure the economic feasibility of the system

3.1.2. Functional Requirement Analysis. In the design of human resource management system based on Java EE, it is necessary to ensure that the system operation is efficient, simple, direct, powerful, and real-time. The general goal of system development is to complete the systematic, standardized, and automatic processing of all kinds of information. Based on the general task of system development, complete the function of human resource management system. It mainly includes organization management, recruitment management, employee information, training, attendance, performance, salary and welfare, enterprise culture, and other management modules. The system example for system administrators is shown in Figure 2, and the system example for common employees is shown in Figure 3.

3.1.3. Analysis of Nonfunctional Requirements. In the design of the human resource management system, nonfunctional requirement design includes the following two points.

(1) Performance requirements of system operation speed, response efficiency, result accuracy, and other aspects

(2) Reliability of users in terms of software failure frequency, easy recovery, severity, and predictability and security requirements to ensure that users use system identity, authorization, and privacy. Ensure safe and reliable operating environment of software system. Ensure that the operating interface of the system is aesthetically available. Ensure that the user's software is scalable, configurable, portable, and maintainable

3.2. Overall Architecture of System Design. Based on MVC three-tier architecture development platform, the system in this paper is divided into three levels, namely, interaction layer, application layer, and data layer. In the three-tier architecture design, the client can only provide better device application services. It has better system development architecture security than other development methods. Users can also access the data layer through the application layer, which effectively improves the overall data security. The architecture diagram of the system is shown in Figure 4.

3.2.1. Interaction Layer. When designing the interactive layer of the human resource management system in this paper, the C# language program is used to design the interactive interface, and the HTML5 technology is used to make forms. It can ensure the human resource management system to provide adaptive functions, combined with differentiated screen size, screen width, and height adjustment. It can also adjust the operation position of the system interface according to the user's requirements.

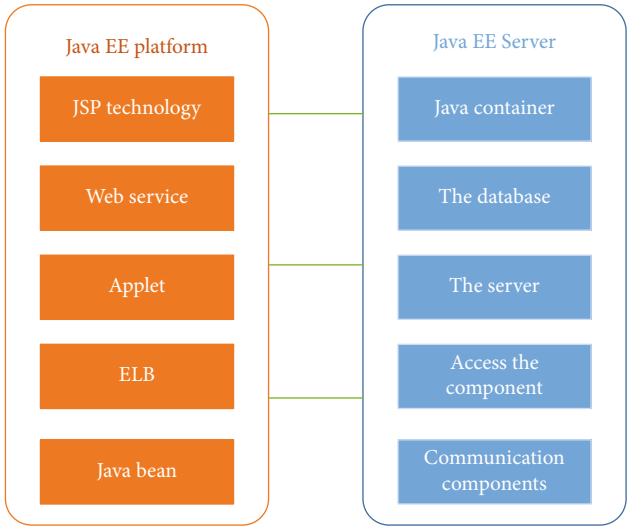

Figure 5: Composition of Java EE technology services.

Table 1: The results of running time.

Data size	k value	k -means	Literature [18]	Proposed	∂_1	∂_2
3×10^7	20	532 s	263 s	257 s	0.538	0.035
	50	1275s	437 s	308 s	0.776	0.314
	100	2886 s	664 s	346 s	0.894	0.498
5×10^7	20	815 s	379 s	328 s	0.616	0.149
	50	2032s	670 s	441 s	0.798	0.358
	100	4610 s	883 s	522 s	0.9	0.424

Table 2: The results of the sum of squares of errors with different algorithms.

Data size	k value	k-means	Literature [18]	Proposed
3×10^7	20	2.487×10^{15}	2.534×10^{15}	2.315×10^{15}
	50	1.018×10^{15}	1.015×10^{15}	1.012×10^{15}
	100	5.126×10^{14}	5.111×10^{14}	5.044×10^{14}
5×10^7	20	6.578×10^{15}	6.603×10^{15}	6.534×10^{15}
	50	2.751×10^{15}	2.762×10^{15}	2.741×10^{15}
	100	1.397×10^{15}	1.396×10^{15}	1.388×10^{15}

3.2.2. The Application Layer. As a large-scale software framework, human resource management system realizes the integration of multiple systems and improves the overall technical compatibility of the system. View the application layer as a development factory pattern, compatible with all subsystem functions. At the same time, the Web server can be utilized to parse the request of the business system, and the corresponding business program can be provided and operated.

3.2.3. The Data Layer. Introduce SQL advanced database technology into the data layer. The establishment of SQL database can realize the effective connection of various functional components and ensure the processing performance of the database and data communication effectiveness. At the same time, it can be processed offline. The data layer wants to encapsulate data by transforming the data business into managed storage statements. Add information processing, expansion, separation, independence, and other functions to ensure that the operation portability of the system is fully improved. This makes it easier for more users to successfully connect to the system.

3.3. Key Service Functions of the System. Considering the actual human resource management needs, the Java EE framework is adopted in the process of human resource management system. It includes user management, employee information, organization, attendance, salary and welfare, performance, recruitment, training, and other different management modules, respectively. The user operation rights of each module are different.

3.3.1. Main Functional Modules

(1) In the login management function module, the user enters the corresponding user name and password after successfully entering the login interface. If the match is successful, you can enter the system. If the match fails, the system displays a message indicating that the user name or password is incorrect and refreshes the login page again

(2) In the attendance management function module, you can log in the attendance-related information of all employees of the enterprise. In this module to achieve the search, add, modify, delete, and other functions

(3) In the enterprise internal transfer management module, you can choose the object of personnel transfer. It can submit the transfer process and complete all levels of approval. It can coordinate with the salary management module to complete the corresponding salary adjustment

(4) In the performance management module, make equations of the corresponding performance appraisal plan. It can set clear assessment object and target, and choose appropriate assessment method and content. In the process of assessment, factors such as assessment principles, standards, methods, candidates, and data collection should be considered comprehensively

(5) In the functional module of salary management, it can be adjusted appropriately according to the changing situation. You can also click Delete to complete the deletion

(6) In the recruitment management function module, fill in the corresponding job information through recruitment. After completing the registration form, you can upload it to the system successfully

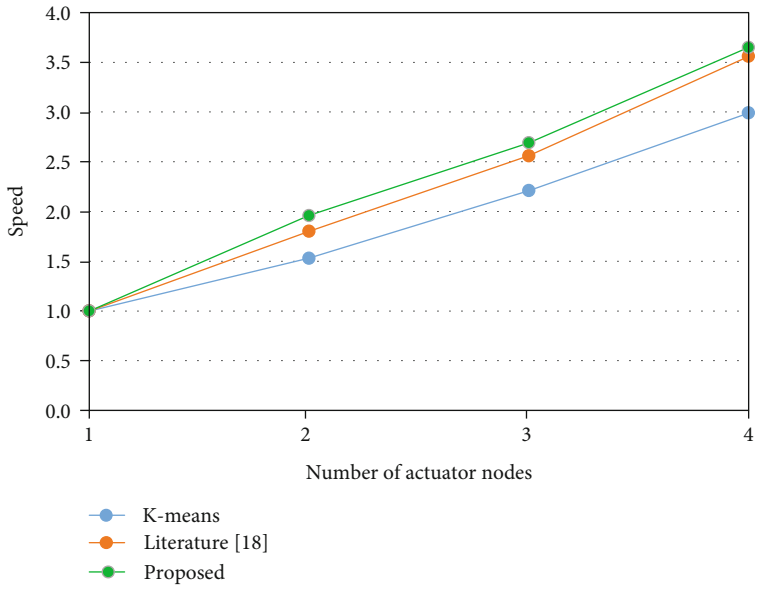

Figure 6: Comparison of parallelization time ($k = 100$).

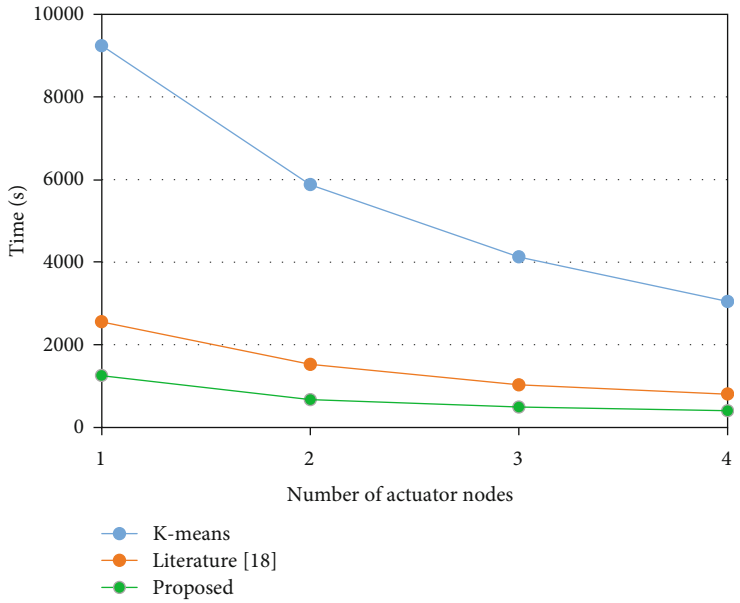

Figure 7: Comparison of algorithm acceleration ratio ($k = 100$).

3.3.2. Database Design. The system is designed to establish SQL database, which can be optimized by SQL statement format. Get uniform specification code according to the object name of the database. It follows the debugging code specification, ensures the good design of database, and comprehensively improves the overall programming calculation efficiency. It can reduce unnecessary data redundancy to some extent and improve the database running efficiency of the system.

In order to realize the data function dynamically, it is necessary to establish the effective connection between the foreground and background of the system, as well as the effective connection between the database and the system code. Establish Java database connection by using Java EE technology. It can provide standard database interface, and the system database connection steps are as follows.

(1) Load the Java EE driver

(2) Provide the main URL according to Java EE

(3) Create a database connection

(4) Create a statement

(5) Execute database SQL statements

(6) Obtain specific processing results

(7) Close the Java EE object

Figure 5 designs Java EE technology service composition for the database of the system.

4. Experiment and Analysis

The data set used in the test is the human resource data set of an enterprise. The writing language used in this experiment is Scala. To verify the effectiveness of the proposed algorithm, traditional k-means algorithm and reference [18] based on Spark framework are selected as the comparison algorithm. The algorithm in reference [18] adopts MLlib L.6.2. The initial cluster center was selected by random selection. All experiments were run for 20 times and averaged.

4.1. Experimental Environment. Spark distributed cluster is adopted in the experiment. Hadoop and Spark are installed on five VMS. Among them, one is responsible for the operation and management of driver programs, and the other four serve as actuator nodes.

Software configuration: Hadoop 2.6.0, JDK 1.7, Spark 1.6.0, MLlib 1.6.2, Scala 2.10.5.

Hardware configuration: 16 G memory, 1024 G hard disk.

4.2. Analysis of Experimental Results

4.2.1. Comparison of Algorithm Performance. Table 1 lists the comparison of the running time of the algorithm in this paper with traditional k-means and reference [18]. Table 2 lists the sum of squares of errors comparisons of the algorithms. Among them, the speed increase ∂_1 and ∂_2 are calculated as follows.

$$\partial_1 = \left(1 - \frac{n_2}{n_0}\right) \times 100\%, \tag{5}$$

$$\partial_2 = \left(1 - \frac{n_2}{n_1}\right) \times 100\%. \tag{6}$$

n_0 is the running time of k-means algorithm. n_1 is the running time of the algorithm in reference [18]. n_2 is the running time of the algorithm in this paper.

As can be seen from the experimental results, the operating efficiency of the algorithm in this paper is significantly improved compared with the traditional k-means and literature [18] algorithm. When k value is small, the speed improvement is relatively insignificant because the improvement strategy in literature [18] has been able to avoid most redundant calculations. However, with the increase of k value, the improvement effect of the algorithm in this paper becomes more obvious.

By comparing the sum of squares of errors, the algorithm in this paper and the algorithm in literature [18] have no negative impact on the clustering quality of the original algorithm.

4.2.2. Scalability Comparison. 4×10^7 data samples were utilized to test the scalability of the algorithm. Figure 6 shows the comparison of parallelization time between traditional k-means, literature [18], and the algorithm in this paper. The algorithm in this paper has a more efficient clustering speed. The running time of the proposed algorithm decreases with the increase of actuator nodes. Meanwhile, due to the time cost of the Spark cluster, the running time of the algorithm does not decrease linearly with the increase of nodes.

Figure 7 shows the acceleration ratio comparison of the algorithms. The proposed algorithm has good scalability. With the expansion of cluster size, the acceleration ratio of the algorithm is basically consistent with that of literature [18].

5. Conclusion

In order to improve the current situation of human resource management, this paper puts forward a design method of human resource management system. Aiming at solving the problem of high computational complexity of traditional k-means algorithm, considering the spatial location relationship between data points and clustering centres and the advantages of grid division, this paper designs a clustering optimization algorithm to save the spatial location information of data points. The comparison results of parallel experiments based on Spark platform show that the computational efficiency of the proposed algorithm is significantly improved, and it has better scalability. On the premise of ensuring the system performance, how to further improve the scalability of the system is the next research direction.

Data Availability

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

Conflicts of Interest

The authors declare no competing interests.

References

[1] S. Inoue and S. Fujita, "Collaborative illustrator with android tablets," in *2019 seventh international symposium on computing and networking workshops (CANDARW)*, pp. 208–214, Nagasaki, Japan, 2019.

[2] V. L. Orlov and E. A. Kurako, "Electronic document management systems and distributed large-scale systems," in *2017 tenth international conference Management of Large-Scale System Development (MLSD)*, Moscow, Russia, 2017.

[3] X. Mai, "Distributed accounting and blockchain technology in financial accounting," *Journal of Physics: Conference Series*, vol. 1881, no. 2, article 022078, 2021.

[4] E. Doychev, P. Malinov, N. Velcheva, and Z. Duchev, "A Genebank architecture: a distributed system for management of plant genetic resources," in *2020 IEEE 10th international conference on intelligent systems (IS)*, pp. 580–583, Varna, Bulgaria, 2020.

[5] I. Makarova, K. Shubenkova, and A. Pashkevich, "Development of an intelligent human resource management system in the era of digitalization and talentism," in *2018 18th international conference on mechatronics-Mechatronika (ME)*, pp. 1–6, Brno, Czech Republic, 2018.

[6] A. Y. Anisimov, A. S. Obukhova, Y. V. Aleksakhina, A. V. Zhaglovskaya, and A. A. Kudra, "Strategic approach to forming a human resource management system in the organization," *International Journal of Economic Perspectives*, vol. 11, no. 2, 2017.

[7] I. Odun-Ayo, S. Misra, N. A. Omoregbe, E. Onibere, Y. Bulama, and R. Damasevicius, "Cloud-based security driven human resource management system," in *ICADIWT*, pp. 96–106, IOS Press, 2017.

[8] C. Chen, K. Li, A. Ouyang, and K. Li, "Flinkcl: an OpenCL-based in-memory computing architecture on heterogeneous CPU-GPU clusters for big data," *IEEE Transactions on Computers*, vol. 67, no. 12, pp. 1765–1779, 2018.

[9] D. Yu, Y. Ying, L. Zhang, C. Liu, and H. Zheng, "Balanced scheduling of distributed workflow tasks based on clustering," *Knowledge-Based Systems*, vol. 199, p. 105930, 2020.

[10] M. Alkathiri, A. Jhummarwala, and M. B. Potdar, "Multi-dimensional geospatial data mining in a distributed environment using Map Reduce," *Journal of Big Data*, vol. 6, no. 1, pp. 1–34, 2019.

[11] T. H. Sardar and Z. Ansari, "Partition based clustering of large datasets using Map Reduce framework: an analysis of recent themes and directions," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 247–261, 2018.

[12] M. M. Fard, T. Thonet, and E. Gaussier, "Deep k-means: jointly clustering with k-means and learning representations," *Pattern Recognition Letters*, vol. 138, pp. 185–192, 2020.

[13] Z. Wang, A. Xu, Z. Zhang, C. Wang, A. Liu, and X. Hu, "The parallelization and optimization of K-means algorithm based on Spark," in *2020 15th International Conference on Computer Science & Education (ICCSE)*, pp. 457–462, IEEE, Delft, Netherlands, 2020, August.

[14] A. S. Chitrakar and S. Petrovic, "Analyzing digital evidence using parallel k-means with triangle inequality on Spark," in *2018 IEEE international conference on big data (big data)*, pp. 3049–3058, IEEE, Seattle, WA, USA, 2018.

[15] M. Saouabi and A. Ezzati, "A comparative between Hadoop MapReduce and Apache Spark on HDFS," in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, pp. 1–4, New York, 2017.

[16] T. Liu, "Personnel matching model of K-means clustering algorithm based on Spark platform," in *Proceedings of the 2020 Conference on Artificial Intelligence and Healthcare*, pp. 143–148, New York, 2020.

[17] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[18] G. Zhang, C. Zhang, and H. Zhang, "Improved K-means algorithm based on density canopy," *Knowledge-Based Systems*, vol. 145, pp. 289–297, 2018.