**Journal of Computing and Electronic Information Management**
ISSN: 2413-1660 | Vol. 17, No. 2, 2025

# Investigation of Enterprise Project Information Management System Integrating Cloud Computing and UML Technology

**Xiaojun Li [a], Jianjuan Fan [b]**

The Smart City Research Institute of China Electronics Technology Group Corporation, Shenzhen, 518000, China
[a] lixiaojun10@cetc.com.cn, [b] fanjianjuan@cetc.com.cn

**Abstract:** The research aims to integrate cloud computing and Unified Modeling Language (UML) technology to optimize enterprise project information management systems. The present work summarizes some recent literature on cloud computing and UML technology and analyzes the functional requirements of enterprise project information management systems. Then, a small financial company's project information management experience is selected as an example for a case study. Besides, the system load balancing algorithm based on Hadoop and the HBase distributed data storage network are constructed through UML software modeling. The experimental results indicate that the system running time span of the information management system built based on cloud computing and UML technology attains 2,750ms under 400 tasks, higher than other algorithms. Meanwhile, the query time of the HBase distributed database network can improve the data transmission efficiency to a certain extent. This research has practical reference and application value in enhancing the efficiency of enterprise project information search and management.

**Keywords:** Cloud computing; UML; Enterprise project information; Optimal designing of systems.

## 1. Introduction

Under the rapid development of information technology in modern society, emerging technologies, such as the Internet, big data, and cloud computing, have progressed rapidly. Correspondingly, enterprise informatization has become a vital measure to improve the efficiency of project information management [1-3]. The continuous excavation of market vitality by reform policies has led to the constant growth of economic dynamism. Due to the integration of various information technologies in traditional enterprise projects, the total amount of project data tends to become more complex and diversified. Managing enterprise project information well and improving the efficiency of data queries while ensuring data security is the focus and top issues of current researchers.

There are many problems in traditional enterprise project management, such as low efficiency and error-prone query results. With the development of data industrialization of information technology [4-6], distributed computing methods provide a new solution to project management problems [7]. Unified Modeling Language (UML) can provide convenience for people to understand software requirements and architecture. Therefore, the design and application of a project information management system based on cloud computing and UML is an inevitable trend of enterprise project management and maintenance in the future [8-10].

Combined with the current development trend of enterprise project management, the present work designs an enterprise project information management system based on cloud computing and UML technology by analyzing the related technical concepts. First, this paper explores the data business process of enterprise project information and summarizes the functional and non-functional requirements of the system. Besides, the Radio Frequency Identification data collection technology of the Internet of Things (IoT) is adopted for data analysis to improve the accuracy of data collection. The main innovation of the research lies in the sorting of system files based on the Hadoop file merging strategy and the storage of system data using the nearest neighbor location matching algorithm. The research has practical reference and application value for the data-based intelligent storage of enterprise project information.

## 2. Recent related work

### 2.1. Cloud Computing and UML Software Modeling Technology

Cloud computing refers to the massive decomposing of data into countless small programs by computer processing programs through the network cloud and then using a distributed system composed of multiple servers for data analysis and processing. Buyya et al. (2018) [11] published a profession of next-generation cloud computing to examine computer science vision. They reported that the cloud computing paradigm had encouraged the creation of scalable global enterprise applications. The findings also suggested that cost correlated better with value for scientific and high-performance computing applications. The authors believed that recent technological developments and paradigms such as serverless computing, software-defined networking, IoT, and processing at network edge created the need for application models designed to address scalability and security. The proposed manifesto could provide research directions for the next decade. Dang et al. (2019) [12] conducted a survey and research on the IoT and cloud computing healthcare systems to establish a seamless data exchange network between systems by connecting intelligent objects and sensors. The study analyzed the well-known security models in the past to deal with security risks and predict the future development trend of healthcare based on the IoT. The authors demonstrated many challenges hindering the development of IoT and cloud computing in healthcare

and analyzed and summarized related security models to prevent possible security risks. Arunarani et al. (2019) [13] applied task scheduling technology to cloud computing and conducted a comprehensive survey of task scheduling strategies and related indicators suitable for cloud computing environments. They confirmed which features should be included and which ones should be ignored in a given system by studying different schedulers. The study identified future research questions related to cloud-based scheduling. Sun et al. (2020) [14] discussed and researched the security and privacy protection of cloud computing, introduced some privacy security risks of cloud computing, and proposed a comprehensive privacy security protection framework. They also conducted a comparative analysis of typical schemes' characteristics and application scope through access control and research on encryption technology based on ciphertext policy attributes. They thought this research could offer achievable technical solutions for security and privacy protection in future cloud computing. Mishra (2020) et al. [15] researched load balancing and request scheduling in cloud computing, reviewed modern load balancing algorithms, and put forward a cloud system architecture to explain cloud systems. Furthermore, the authors compared performance parameters between different studies of load balancing in the cloud. They believed this research had practical reference value for the related research on cloud computing architecture and load balancing algorithms in the future. Bello et al. (2021) [16] investigated cloud computing and related use cases in the construction industry, introduced the existing cloud computing applications in construction, and discussed the future potential of cloud computing in the construction industry. They stated that cloud computing should be more widely adopted in the construction industry. They highlighted the obstacles and solutions to the broader adoption of cloud computing in the construction industry. Sadeeq et al. (2021) [17] investigated the issues and challenges of IoT and cloud computing. The findings suggested that it was unwise to store all raw data locally in IoT devices with the exponential growth of industrial IoT data.

The use of UML software modeling technology for enterprise project information management has become a new trend in the development of computational information technology. For example, Khaiter et al. (2019) [18] used UML software modeling techniques to study the environmental software modeling framework for conceptualizing sustainable management and discuss the overall multi-layer architecture of the framework and its key software components. The authors explained the mechanisms of practical applications of sustainability so that all goods and services provided by ecosystems were fully quantified, assessed, and incorporated into decision-making. Mumtaz (2019) [19] studied the UML model framework for software refactoring, introduced the latest research on detecting unpleasant odors in UML models, and compared and evaluated odor detection techniques using the proposed evaluation framework. The study showed that lousy smell detection in class and sequence diagrams was completed through design patterns, software indicators, and predefined rules. Besides, the use of metrics and predefined rules was helpful to detect the model smell in use cases. Ozkaya et al. (2020) [20] surveyed utilizing UML software from different software architecture viewpoints and 109 practitioners with different profiles. The survey results indicated that UML's class diagram was the first choice for practitioners. Functional structure, data structure, concurrency structure, software code structure, system installation, etc., together constituted a complete software architecture. Munthe et al. (2020) [21] conducted UML modeling and software black-box testing for a school payment information system. They found that the black-box evaluation method could be used as software evaluation. In addition, external control mechanisms could adequately control the information system software when system testers tested the school payment information system as designed and focused. Abdalazeim et al. (2021) [22] used an objective UML model and domain ontology to generate natural language requirements. Their research showed that existing research methods had certain robustness and limitations. Triandini et al. (2021) [23] used UML software to measure the similarity of graphs and introduced the results of similarity measurement between UML graphs of different software products. The study showed that the new algorithm could be tested by compiling the calculation information of each graph, adjusting the calculation method of semantic and structural similarity, and constructing the data test set.

## 2.2. The Enterprise Project Information Management System

The system theory has practical application value for improving enterprise operation efficiency in the process of enterprise project information management. For instance, Gorkhali et al. (2019) [24] explained information system research from a system theory perspective when researching enterprise architecture and information management systems. They summarized all the latest research advances in systems theory in three areas: enterprise systems, enterprise information systems, and enterprise architecture. The research results indicated that system theory and system perspective were critical research methods for enterprise information management. Tian et al. (2020) [25] studied the information system design of IoT-enabled manufacturing enterprises and optimized the production information management system and workshop dynamic scheduling strategy based on the industrial IoT technology. The computational experiments suggested that the values of relative error and average ideal distance were 11.3 and 13.74, respectively, demonstrating the feasibility and efficiency of the proposed scheduling method. Ye et al. (2020) [26] studied the intelligent physical system for enterprise information strategic planning and management. They realized the online scheduling optimization of enterprise data building energy management in the innovative grid environment. They proved that the proposed power prediction method could provide real-time input to consumers and promote the better application of power in the intelligent physical system. Cui et al. (2021) [27] investigated the anonymous proof mechanism in enterprise information management and proposed a novel direct unknown authentication scheme by introducing short signatures in enterprise information management. The proposed plan simplified the signing and verification process of member certificates and reduced its computational cost to meet trusted platform modules' normative and security requirements. Zdravković et al. (2021) [28] optimized artificial intelligence-based manufacturing enterprise information systems and reviewed artificial intelligence applications in product life cycle management and human resources. They believed that this research had practical reference value for manufacturing enterprises' digital and intelligent control. Kim (2021) [29] focused on corporate

information security portals of financial companies. The scholars established and operated information security systems, checked system vulnerabilities, and protected information privacy through systematic design and implementation. This research had significant technical reference value for the information protection governance and security management of financial companies.

To sum up, with the development of large-scale and complex enterprise engineering projects, traditional information processing solutions have been difficult to meet the needs of engineering project management. Therefore, the project management system based on cloud computing technology has gradually become the research focus of related scholars. The system archiving of project information based on the project management features combined with the project management information architecture can realize the high-speed sharing of associated resources.

# 3. Research methods

## 3.1. Functional Requirements of the Information Management System

The project management requirements are the priority when designing an enterprise project management platform based on cloud computing technology, including project application requirements, project initiation requirements, and project closure requirements [30]. Firstly, the qualified employees submit the required project application through a specific platform. Then, the application is approved by the leader of the project department. After the approval, the project is initiated. When a project is approved, the project name, person in charge, responsible department, and project type need to be filled in completely. In addition, the completed projects need to submit the project closing application through the platform and complete the project closing operation after being approved by the leader of the project department. Furthermore, the system needs to manage and maintain the information of all personnel and departments in the company's projects. The platform needs a cloud computing storage environment that supports massive data storage. Figure 1 displays the functional requirements of the information management system.

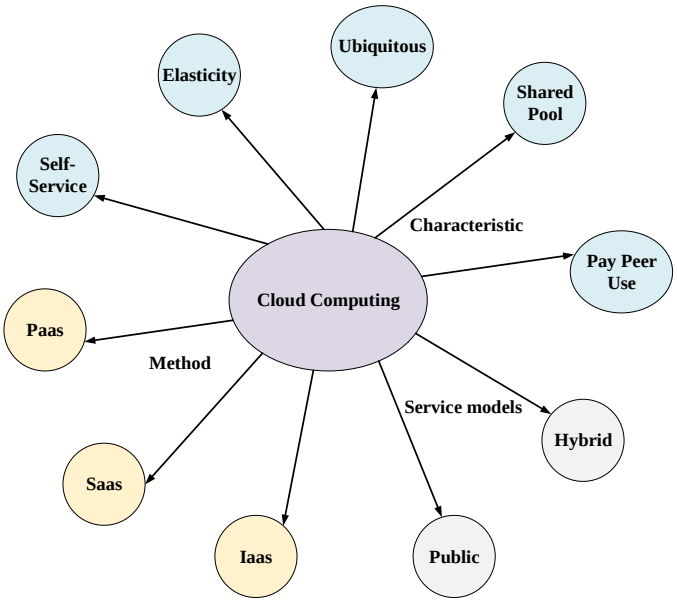


**Figure 1.** Functional requirements analysis of the information management system

## 3.2. Data Input for the Information Management System

For each data recorded by the management system, the value of $X$ may be $x_i(i = 1,2, \dots \dots n)$. Eq. (1) indicates the information entropy recorded.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \quad (1)$$

In Eq. (1), $p(x_i)$ represents the probability of $X = x_i$, and $n$ denotes the number of possible values of $X$. The corresponding information entropy of the classified attribute A of all data in the system record can be calculated according to Eq. (2).

$$E(A) = -\sum_{j=1}^{n} \frac{|x_j|}{|x|} * H(X_j) \quad (2)$$

In Eq. (2), $H(X_j)$ denotes the information entropy of the data when $X = X_j$. In addition, the information gain is the difference between Eq. (1) and Eq. (2), as presented in Eq. (3).

$$\mathrm{G} = \mathrm{H(X)} - \mathrm{E(A)} \quad (3)$$

Then, the information gain rate of the system data in terms of attribute A can be described as Eq. (4).

$$\mathrm{R(A)} = \frac{\mathrm{C(A)}}{\mathrm{Splitinfo(A)}} \quad (4)$$

In Eq. (4), SplitInfo (A) represents the information decision factor of attribute A, which can be written as Eq. (5).

$$\text{SplitInfo (A)} = -\sum_{j=1}^{n} \frac{|x_i|}{x} * \log_2 \frac{|x_1|}{x} \quad (5)$$

According to the normal distribution empirical function, the time matching degree of the system can be recorded as:

$$\mathrm{N}(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\Delta t - t - \mu)^2}{2\sigma^2}} \quad (6)$$

The time matching degree of all data is normalized and simplified according to the characteristics of the normal distribution. The time matching degree is finally defined as Eq. (7).

$$N(t) = \begin{cases} 1 & \Delta t \le t \\ e^{-\pi(\Delta t - t)^2} & \Delta t > t \end{cases} \quad (7)$$

In Eq. (7), $\Delta t$ represents the time interval. The time matching degree of the data stored at the moment $t_{i+1}$ needs to meet Eq. (8).

$$\eta(t_{i+1}) = max\big(H(t_i) \times H^T(t_{i+l})\big) \quad (8)$$

In Eq. (8), $H(t_i)$ represents the distance matching degree of the data stored at the moment $t_i$, and $H^T(t_{i+l})$ refers to the distance matching degree of the stored data at the moment $t_{i+l}$.

Eq. (9) describes the comprehensive matching degree between the current storage location and the candidate storage location.

$$\eta = (\omega_d d + \omega_\alpha \alpha)c \quad (9)$$

In Eq. (9), $\omega_\alpha$ and $\omega_d$ represent the distance between the current storage location and the candidate storage location and the data storage center, respectively; $d$ and $\alpha$ signify the distance coefficient; $c$ stands for the connection coefficient between the storage locations.

The normalized value $\bar{d}_i$ and floating value $\bar{\alpha}_i$ of the stored data are calculated according to:

$$\bar{d}_i = 1/(1 + d_i/\Delta_{GPS}) \quad (10)$$

$$\bar{\alpha}_i = 1/(1 + \alpha_i/\Delta\alpha) \quad (11)$$

Eq. (12) indicates the matching degree between the anchor point and the alternative storage location.

$$\eta_i^r = \big(\omega_d \bar{d}_i^r + \omega_\alpha \bar{\alpha}_i^r\big)c_i^r \quad (12)$$

In Eq. (12), $c_i^r$ represents the maximum value of the connectivity between the alternative storage location and the storage location at the moment $t_i$.

For the distance between the storage location and the data reader, the matching speed for reading data is recorded as:

$$\eta_i^r = \delta_i^r c_i^r \tag{13}$$

Figure 2 reveals the data information entry frame of the information management system at this time.

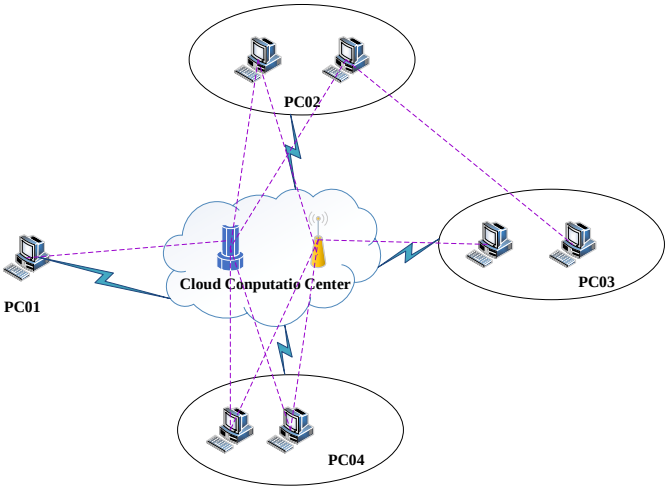


**Figure 2.** Frame of data information entry of the information management system

According to the feedback principle, the same data is stored at the next moment. The matching speed of data storage can be recorded as:

$$\eta_{i+1}^r = \eta_i^r (\delta_{i+1}^r c_{i+1}^r) \tag{14}$$

At this time, considering the factors of matching degree feedback and the set of matching degree algorithms, the matching degree at the moment $t_{i+1}$ can be expressed as Eq. (15).

$$H'_{i+1} = H_i H_{i+1}^T \tag{15}$$

In addition, $H_i$ is the matching degree set at the current moment for the delay matching algorithm. Then, the matching degree set at the next moment is expressed as Eq. (16).

$$H^t = H^{t-1}\left(\omega_d \bar{D}^t + \omega_\alpha \dot{\mathrm{A}}^t\right) C_{r,t}^T \tag{16}$$

Then, the time complexity of the system processing stored data is analyzed. The total data processing time $T$ can be expressed as Eq. (17).

$$T = \sum_{i=1}^n t_{m,i} + \sum_{i=1}^n t_p + \sum_{i=1}^n t_{r,i} \tag{17}$$

In Eq. (17), $t_{m,i}$, $t_p$ and $t_{r,i}$ represent the time spent in data search, intermediate data processing, and data screening, respectively. Since the time of each calculation of $t_{m,i}$, $t_p$ and $t_{r,i}$ is roughly the same, the time spent in a single data processing is the simplified as Eq. (18).

$$T_i = max\{t_{m,i}\} + max\{t_{r,i}\} + t_p \tag{18}$$

In addition, the time for data entry into the storage system is divided into two parts. The time spent passing through each storage unit is expressed as $\frac{d_{r_i}}{v_{r_i}}$, and the time spent passing through the central processing unit is denoted as $t_i$.

$$t_{r_i} = t_i + \frac{d_{r_i}}{v_{r_i}} \tag{19}$$

The time interval of data entry is calculated according to Eq. (20).

$$t_{r_i} = \begin{cases} t_i & (r_i = r_s) \\ t_i + \dfrac{d_{r_i}}{v_{r_i}} & (r_i \neq r_s \text{ or } r_e) \\ t_{i+1} & (r_i = r_e) \end{cases} \tag{20}$$

Eq. (21) indicates the average read speed of the system disk.

$$V_s = nL / \sum_{i=1}^n t_i \tag{21}$$

In Eq. (21), $V_s$ represents the average access time of the system, $t_i$ denotes the time it takes for the $i$-th data to be stored in the system, $n$ refers to the total amount of data, and $L$ stands for the storage bandwidth. The instantaneous access speed of the system is presented as:

$$V_t = \sum_{k=1}^m v_k / m \tag{22}$$

$$Q = 60n/(t_1 - t_0) \tag{23}$$

where $V_t$ represents the instantaneous access speed, $v_k$ refers to the instantaneous access speed when the $k$-th data is stored, and $k$ signifies the total amount of data access. The total amount of data access is systematically judged. The structural frame is shown in Figure 3.

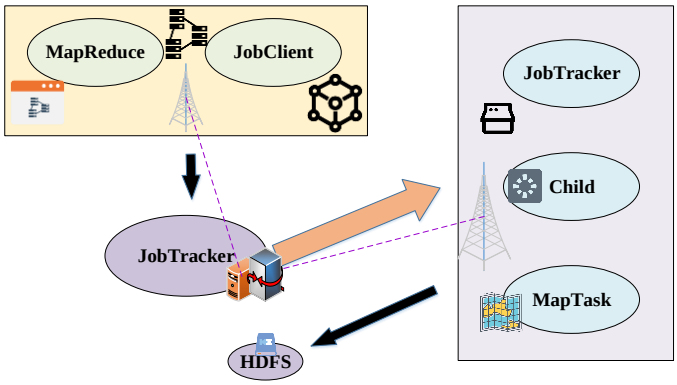


**Figure 3.** Overall structural framework of the information management system

## 3.3. Information Storage and Query of the Warehouse Information Management System for Enterprises

For the adjacent position of the stored data, the position weight of the currently stored data and the rest of the data is set by the Euclidean distance formula. The adjacent position matching is shown in Eq. (24) ~ Eq. (27).

$$d_i = \sqrt{\sum_{i=1}^n \left(rssi_l^i - rssi_l^j\right)^2} \tag{24}$$

$$w_i = \frac{1/d_i}{\sum_{i=1}^k 1/d_i} \tag{25}$$

$$\sum_{i=1}^k w_i = 1 \tag{26}$$

$$(\hat{x}, \hat{y}) = \sum_{i=1}^k w_i (x_i, y_i) \tag{27}$$

Among them, $d_i$ denotes the Euclidean distance, $rssi_l^i$ represents the Received Signal Strength Indication (RSSI) position information of the $i$-th reference point received by the $l$-th data receiver, and $rssi_l^j$ refers to the RSSI position information of the $j$-th reference point received by the $l$-th data receiver. Besides, $w_i$ stands for the distance weight of the nearest neighbor point $i$, $(\hat{x}, \hat{y})$ represents the position coordinate of the fixed point $i$, $(x_i, y_i)$ denotes the position coordinate of the fixed point $i$ stored in the database, and $K$ is the total amount of data.

The distance discrimination between the data position to be processed and the known data position is obtained through the cosine value of the signal strength vector of the two positions, as shown in Eq. (28).

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n x_i \cdot y_i}{\left(\sum_{i=1}^n x_i^2 \cdot \sum_{i=1}^n y_i^2\right)^{1/2}} \tag{28}$$

In Eq. (28), $A$ represents the average signal strength of the point to be detected after being processed by $n$ signal processors, $B$ refers to the average coordinate position in the database. After calculation, the coordinate data closest to the average place in the database is selected as the data storage

location.

In addition, when locating data in the system, since the system noise interference will interfere with the strength of the collected data, the Gaussian filtering formula is used to eliminate the interference. The information with a success probability greater than 90% is selected.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} rssi_i \tag{29}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (rssi_i - \mu)^2} \tag{30}$$

In Eq. (29) and Eq. (30), $\mu$ represents the mean value of the data signal strength RSSI, $n$ refers to the number of data, and $\sigma$ indicates the standard deviation of RSSI data. The data signal strength is strengthened After the above Gaussian filter processing. Figure 4 illustrates the data signal processing flow at this time.

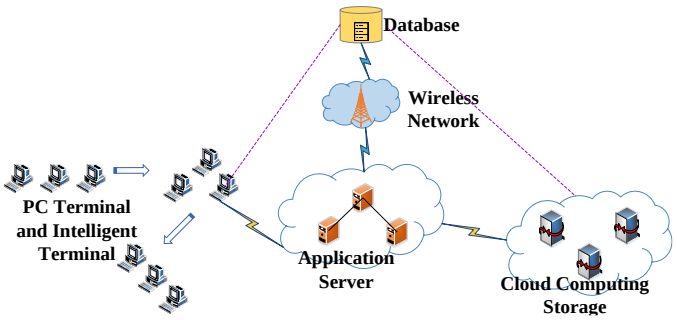


**Figure 4.** Process flow of enterprise warehousing data information

In addition, mean filtering is performed on the data to further improve the accuracy of the stored data, as shown in Eq. (31).

$$RSSI_{avg} = \frac{1}{n}\sum_{i=1}^{n} rSSi_i \tag{31}$$

In Eq. (31), $RSSI_{avg}$ represents the mean value of the data signal strength, $n$ refers to the number of data, and $rSSi_i$ represents the data signal strength value at the $i$-th position. At the same time, it is necessary to divide the data area of the stored data signal by clustering. First, the clustering distance from the storage object to the data center is calculated according to Eq. (32).

$$D(i,r) = \sqrt{\left(F_i(RSSI_n) - C_r(RSSI_n)\right)^2} \tag{32}$$

In Eq. (32), $D(i,r)$ refers to the Euclidean distance from the first fingerprint object to the data center, $F_i(RSSI_n)$ represents the RSSI value of the $i$-th fixed-point transmission received by the $n$-th data receiver, and $C_r(RSSI_n)$ signifies RSSI sent by the $r$-th data cluster center to the data receiver. According to the minimum Euclidean clustering principle, the data storage area is divided and updated according to Eq. (33).

$$C^r = \frac{1}{N_r}\sum_{t=1}^{N_r} x_t (x_t \in C_r, r = 1,2,\cdots,K) \tag{33}$$

In Eq. (33), $C_r$ represents the data set of the $r$-th area, $C^r$ refers to the data clustering center of the $r$-th area, $N_r$ signifies the number of data elements stored in the $r$-th area, and $K$ is the total number of all regions.

In addition, all data storage areas are arranged according to the size of the Euclidean distance. K reference points are selected, and the weight assigned to each reference point is $\omega_j$, which is calculated according to Eq. (34).

$$\omega_j = \frac{\frac{1}{D_j}}{\sum_{j-1}^{K}\frac{1}{D_j}} \tag{34}$$

Finally, the stored position coordinates are recalculated according to the position coordinates of $K$ reference points and the corresponding weight values, as shown in Eq. (35).

$$(x,y) = \sum_{j=1}^{K} \omega_j (x_j, y_j) \tag{35}$$

The data collection is set to systematically collect enterprise project information to analyze and compare the performance of the enterprise information management system integrating cloud computing and UML technology. Figure 5 displays the codes of the core Hadoop-based file merging algorithm.

| | |
|---|---|
| 1 | **Algorithm:** |
| 2 | **Input: File, Node name** |
| 3 | **Output: Data stream merging** |
| 4 | **Parameters: Folders, Merged files, Temporary files** |
| 5 | Enter the storage file path. |
| 6 | String [] fileList=File.list(); |
| 7 | **For**(int i=0;i<fileList.length;i++){ |
| 8 | File path re-extraction |
| 9 | **If**(!readfile.isDirectory()){ |
| 10 | Upload the file to the department node |
| 11 | Int temp; |
| 12 | **While**(temp=input.read(buffer)!=-1){ |
| 13 | Output.write(buffer,0,temp); |
| 14 | } |
| 15 | **End While** |
| 16 | **End if** |
| 17 | **End for** |
| 18 | } |
| 19 | } |
| 20 | **The data stream is merged into the file, and the file merging is completed.** |

**Figure 5.** Flow of the core file merging algorithm based on Hadoop

## 3.4. Case Study

This paper designs the enterprise project information management system under the guidance of software engineering theory by studying cloud computing and UML technology. Then, a small financial company is selected as a case to test the information management system's performance. The Hadoop-based information management method [31] is tested by various databases, such as the Hbase distributed database [32]. MapReduce constructs the computing model of the system parallel processing. In addition, for the application server used by the management system, the specific parameters are configured as follows. The operating system is MacOS Sierra 10.12.5, the device model is MacBook Pro (13-inch, 2017, Two Thunderbolt 3 ports), and the CPU adopts 2.3 GHz Intel Core i, the system memory is 8G 2133MHz LPDDR3. The graphics card is set to Intel Iris Plus Graphics 640 1536 MB. Moreover, it is necessary to compare the performance of the Hadoop-based architecture algorithm reported here with some load balancing processing algorithms, such as Met, Mct, Min, Med, and Max, to evaluate the system performance. Figure 6 reveals the experimental test environment for cloud computing and UML software modeling.

**(a)** **(b)**

**Figure 6.** Experimental test environment for software modeling (a. the system environment of cloud computing; b. the system testing environment of UML software modeling)

# 4. Results and discussion

## 4.1. Experimental Results of Load Balancing Algorithms

Different load balancing algorithms are selected to apply to the system load balancing to analyze the information management and storage effect of the system. The performance of the Hadoop-based algorithm proposed in this paper is compared with several load balancing processing algorithms, including Met, Mct, Min, Med, and Max. The results are shown in Figure 7. In addition, regarding the performance comparison between MySQL database and Hbase distributed database, repeated tests are conducted in 3 groups, and the results are presented in Figure 8. Figure 9 compares the test results of these algorithms under different numbers of nodes with MySQL database and Hbase distributed database.

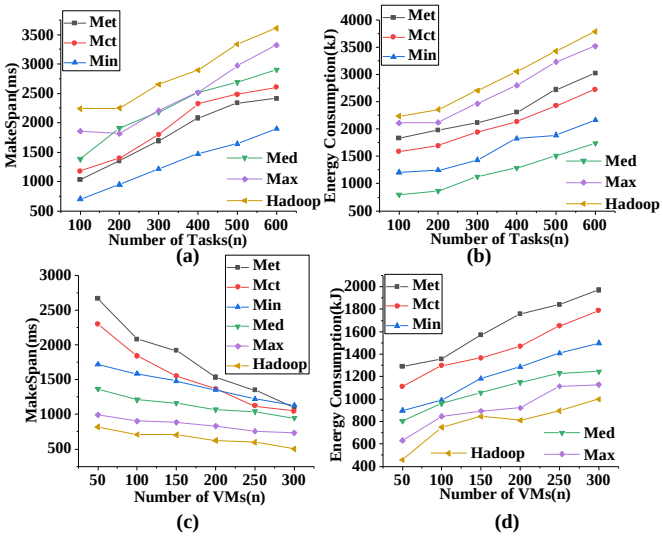


**Figure 7.** System performance test results of load balancing algorithms (a. the change curve of the system running time span with the increase of the storage task amount; b. the change curve of the system running energy consumption with the rise of the storage task amount; c. the change curve of the system running time span with the increase in virtual computer nodes; d. the change curve of the energy consumption of the system operation with the increase in virtual computer nodes)

According to Figure 7(a), the task running time of several load balancing algorithms gradually increases as system storage tasks increase. Compared with the other algorithms, the running time of the Hadoop-based algorithm proposed here varies most obviously with the number of storage tasks. When the task amount is 400, the system running time span reaches 2,750ms, higher than the other algorithms. The change curve of the energy consumption value of system operation in Figure 7(b) suggests that with the increase of the number of tasks, the energy consumption value of the system operation of several algorithms is gradually increasing. Among them, the system construction method based on Hadoop proposed here consumes the most energy. The reason may be the energy dissipation caused by the excessive number of user queries after the system is built. As shown in Figure 7(c), the system running time span gradually decreases with the increase of virtual computer nodes. However, the time span of the Hadoop-based system construction method decreases the fastest, down to around 500ms when the number of nodes is 300. Through Figure 7(d), the energy consumption of the system operation increases gradually with the increase of the number of virtual computer nodes. The method based on Hadoop can reduce the energy loss of the system to a certain extent and achieve better experimental results under low energy consumption than other algorithms.

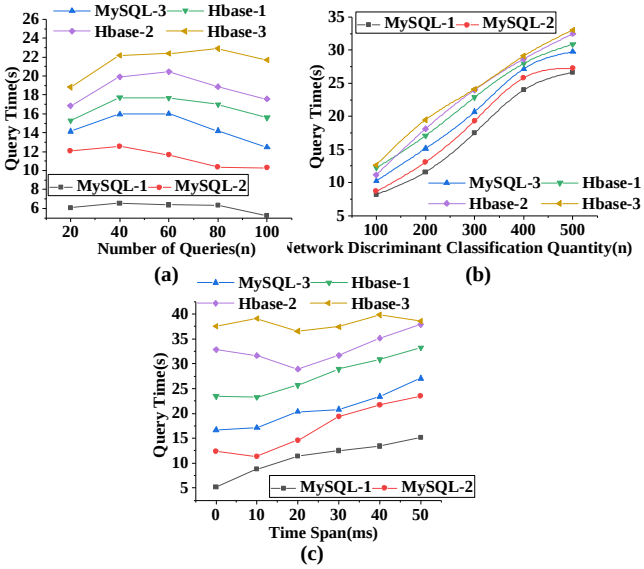


**Figure 8.** Grouping test results of the MySQL database and Hbase distributed database (a. the change curve of query time with the increase of the number of queries; b. the change curve of the query time consumption with the rise in the number of classified queries; c. the change curve of the query time with the increase of the system time span)

As can be seen from Figure 8(a), compared with the MySQL database, the query time of the Hbase distributed database increases more with the increase of the number of system queries. When there are 80 queries, the system query time of several groups of experimental data of Hbase is about 20s on average. Figure 8(b) indicates that with the increase in the number of classified queries, the time-consuming of querying information in the MySQL database and Hbase distributed database is also increasing. However, the Hbase distributed database takes more time than the MySQL database, which also achieves higher accuracy. According to Figure 8(c), with the increase of the system time span, the query time fluctuates, but overall, the query time shows an upward trend. Besides, the query time of the Hbase distributed database is greater than that of the MySQL database, indicating that the MySQL database can improve the efficiency of data transmission.

Figure 9(a) demonstrates that with the increase in the number of nodes for the same database, the time spent in system querying also increases. In general, the system query time when the number of nodes is 30 must be greater than the system query time when the number of nodes is 10. Also, the Hbase database takes more query time than the MySQL database. As can be seen from Figure 9(b), with the increase of the number of query results, the time value spent on filtering data in the MySQL database fluctuates wildly. In contrast, the data query time in the HBase database is relatively stable. According to Figure 9(c), when the number of nodes is 20, the task execution time of the Hbase database is 34s on average. Although it is not the optimal result, the data query result can remain stable for a long time, which can improve the efficiency of data queries and optimize the security management of project information.

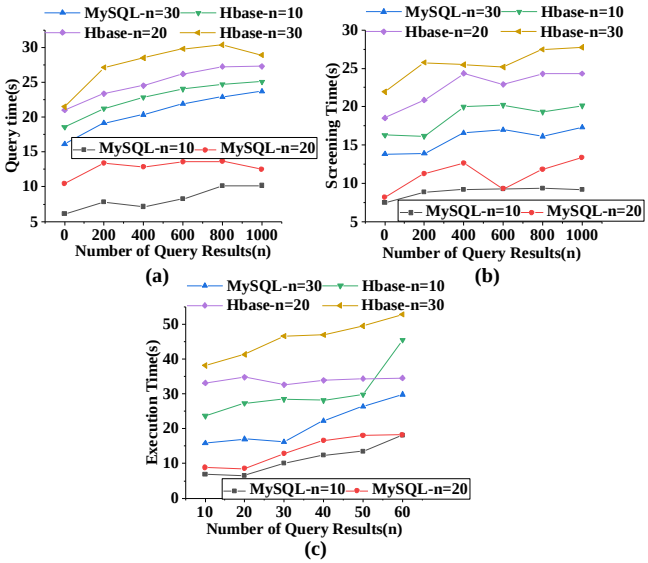


**Figure 9.** Performance comparison of different databases with different numbers of nodes (a. query time consumption curve varying with the number of query results; b. time consumption curve of filtering data as the number of query results increases; c. time consumption curve of executing tasks as the number of query results increases)

## 4.2. Data Processing Performance Comparison of Different Information Management Systems

This section evaluates the data processing performance of information management systems constructed by different methods. The BigTable and MapReduce system construction schemes are selected for performance comparison. Figure 10 provides the change curve of the total number of tasks in the system capacity of different data processing systems and the change curve of the average task reading speed with time. In addition, Figure 11 shows the overall execution time of different system matching schemes.

According to the change curve of the percentage of total tasks in the system capacity of different data processing systems in Figure 10 (a), with the increase of the amount of stored data, the optimization of the system can continuously reduce the percentage of system capacity. Nevertheless, there's a certain amount of decline. When the total amount of data reaches 1,500, the complete tasks of the MapReduce data processing system account for only 7.5% of the system capacity on average. In contrast, the total tasks of the BigTable data processing system account for an average of 15% of the system capacity at this time. Therefore, the optimization performance of the MapReduce data processing system is even better. Figure 10(b) shows the average reading speed variation curve of different data processing systems over time. Specifically, the average reading speed of the BigTable data processing system fluctuates wildly. On the contrary, the reading speed of the MapReduce data processing system is relatively smooth, protecting the disk of the data reading system and prolonging the service life of the system to a certain degree.

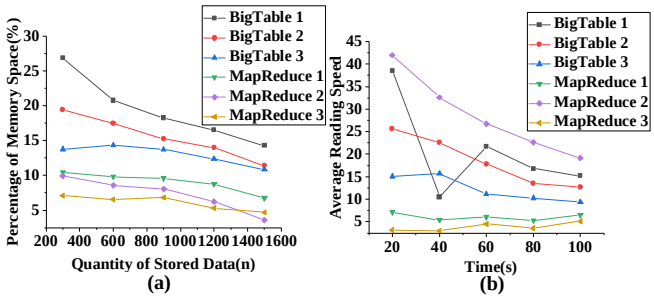


**Figure 10.** Performance comparison results of different data processing systems (a. the change curve of the percentage of total tasks in the system capacity of different data processing systems; b. variation curve of task average reading speed with time in different data processing systems)

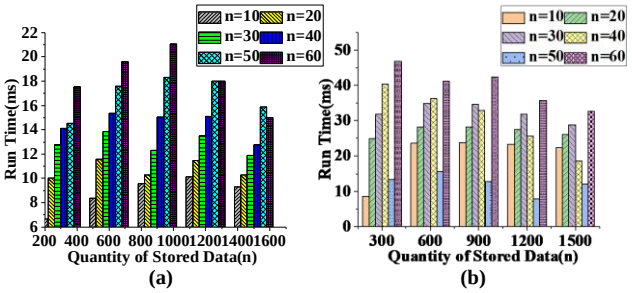


**Figure 11.** Comparison of the execution time of two system matching schemes with different number of nodes (a. comparison of the overall execution time of the BigTable matching scheme; b. comparison of the overall execution time of the MapReduce matching scheme)

From the comparison results of the overall time between the BigTable system and the MapReduce system in Figure 11, with the increase of the number of nodes, the execution time of the matching scheme continues to increase. However, this growth is limited to a certain extent. The actual execution time of the BigTable solution continues to increase when the number of nodes increases from 10 to 60 but decreases to some extent when the total amount of data stored exceeds 1,500. The MapReduce system reaches the maximum execution time when the number of nodes is 40. After that, with the increase of the number of nodes, the execution time of the plan decreases to a certain extent. These results prove that compared with the BigTable system, the MapReduce system reported here is more conducive to the security and stability of project information management.

# 5. Conclusion

With the continuous development of the social economy and technology, enterprise projects tend to be complicated and large-scale. The extensive collection of various information and technologies gradually makes the project a complex long-term system engineering. This paper aims to design and optimize the enterprise project information management system based on cloud computing and UML software modeling technology. The research starts from cloud computing and UML software modeling technology and studies the development process of enterprise project information management systems by sorting out essential concepts and reviewing the literature. Under the guidance of software engineering, this paper analyzes the functional requirements of the information management system. Then, enterprise storage information is taken as an example to simulate the data information entry and information storage location query of the information management system. The

research results show that the system running time span of the information management system built based on cloud computing and UML technology is 2,750ms under 400 tasks, which is higher than the other algorithms. At the same time, the query time of the HBase distributed database can be determined by the efficiency of data transmission to a certain extent. The research has significant reference value for the digital and intelligent development of enterprise project information. However, some shortcomings remain unavoidable. First, some functions of the project information management system designed here need to be further improved. The system needs to be adjusted according to the different types of projects. Besides, system functions need to be perfected in the later research to improve the system. Second, in the cloud computing storage model, the system reported here requires the participation of multiple servers. The follow-up research will concentrate on keeping files as efficient as possible while reducing the cost of hardware facilities.

# References


[1] Kwilinski, A. (2018). Mechanism of formation of industrial enterprise development strategy in the information economy. Virtual Economics, 1(1), 7-25.

[2] Gruden, N., & Stare, A. (2018). The influence of behavioral competencies on project performance. Project Management Journal, 49(3), 98-109.

[3] Putz, B., Dietz, M., Empl, P., & Pernul, G. (2021). Ethertwin: Blockchain-based secure digital twin information management. Information Processing & Management, 58(1), 102425.

[4] Li, Y., Yang, X., Ran, Q., Wu, H., Irfan, M., & Ahmad, M. (2021). Energy structure, digital economy, and carbon emissions: evidence from China. Environmental Science and Pollution Research, 28(45), 64606-64629.

[5] Imamov, M., & Semenikhina, N. (2021). The impact of the digital revolution on the global economy. Linguistics and Culture Review, 5(S4), 968-987.

[6] Javed, A. (2020). The scope of information and communication technology enabled services in Promoting Pakistan Economy. Asian Journal of Economics, Finance and Management, 1-9.

[7] Bogdanova, M., Parashkevova, E., & Stoyanova, M. (2020). Agile project management in governmental organizations–methodological issues. International E-Journal of Advances in Social Sciences, 6(16), 262-275.

[8] Zhou, C., & Cao, Q. (2019). Design and implementation of intelligent manufacturing project management system based on bill of material. Cluster computing, 22(4), 8647-8655.

[9] Qin, Q. (2018). Design of project management system based on web technology. Information Management and Computer Science (IMCS), 1(1), 6-8.

[10] Cheng, Z., Lin, M., Ma, Y., & Fan, C. (2018). An Engineering Project Management System Design of Fengyun Meteorological Satellite. Information Systems and Signal Processing Journal, 3(1), 1-4.

[11] Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., ... & Shen, H. (2018). A manifesto for future generation cloud computing: Research directions for the next decade. ACM computing surveys (CSUR), 51(5), 1-38.

[12] Dang, L. M., Piran, M., Han, D., Min, K., & Moon, H. (2019). A survey on internet of things and cloud computing for healthcare. Electronics, 8(7), 768.

[13] Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. Future Generation Computer Systems, 91, 407-415.

[14] Sun, P. (2020). Security and privacy protection in cloud computing: Discussions and challenges. Journal of Network and Computer Applications, 160, 102642.

[15] Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. Journal of King Saud University-Computer and Information Sciences, 32(2), 149-158.

[16] Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. Automation in Construction, 122, 103441.

[17] Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. Qubahan Academic Journal, 1(2), 1-7.

[18] Khaiter, P. A., & Erechtchoukova, M. G. (2019). Conceptualizing an Environmental Software Modeling Framework for Sustainable Management Using UML. Journal of Environmental Informatics, 34(2).

[19] Mumtaz, H., Alshayeb, M., Mahmood, S., & Niazi, M. (2019). A survey on UML model smells detection techniques for software refactoring. Journal of Software: Evolution and Process, 31(3), e2154.

[20] Ozkaya, M., & Erata, F. (2020). A survey on the practical use of UML for different software architecture viewpoints. Information and Software Technology, 121, 106275.

[21] Munthe, I. R., Rambe, B. H., Pane, R., Irmayani, D., & Nasution, M. (2020). UML Modeling and Black Box Testing Methods in the School Payment Information System. Jurnal Mantik, 4(3), 1634-1640.

[22] Abdalazeim, A., & Meziane, F. (2021). A review of the generation of requirements specification in natural language using objects UML models and domain ontology. Procedia Computer Science, 189, 328-334.

[23] Triandini , E., Fauzan , R., Siahaan , DO, Rochimah , S., Suardika , IG, & Karolita , D. (2021). Software similarity measurements using UML diagrams: A systematic literature review. Register: Journal Scientific Technology System Information , 8 (1), 10-23.

[24] Gorkhali, A., & Xu, L. D. (2019). Enterprise architecture, enterprise information systems and enterprise integration: A review based on systems theory perspective. Journal of Industrial Integration and Management, 4(02), 1950001.

[25] Tian, S., Wang, T., Zhang, L., & Wu, X. (2020). The Internet of Things enabled manufacturing enterprise information system design and shop floor dynamic scheduling optimisation. Enterprise Information Systems, 14(9-10), 1238-1263.

[26] Ye, C., Song, X., Vivekananda, G. N., & Savitha, V. (2021). Intelligent physical systems for strategic planning and management of enterprise information. Peer-to-Peer Networking and Applications, 14(4), 2501-2510.

[27] Cui, C., Li, F., Li, T., Yu, J., Ge, R., & Liu, H. (2021). Research on direct anonymous attestation mechanism in enterprise information management. Enterprise Information Systems, 15(4), 513-529.

[28] Zdravković, M., Panetto, H., & Weichhart, G. (2021). AI-enabled Enterprise Information Systems for Manufacturing. Enterprise Information Systems, 1-53.

[29] Kim, D. H. (2021). Design and Implementation of Enterprise Information Security Portal (EISP) System for Financial Companies. Convergence Security Journal, 21(1), 101-106.

[30] Hryshko, V., & Kryvenko, O. (2018). Efficiency of project management of the enterprise development under current economic conditions. Економіка і регіон, (3), 65-70.

[31] Wu, W., Lin, W., Hsu, C. H., & He, L. (2018). Energy-efficient hadoop for big data analytics and computing: A systematic review and research insights. Future Generation Computer Systems, 86, 1351-1367.

[32] Kalogiannis, S., Deltouzos, K., Zacharaki, E. I., Vasilakis, A., Moustakas, K., Ellul, J., & Megalooikonomou, V. (2019). Integrating an openEHR-based personalized virtual model for the ageing population within HBase. BMC medical informatics and decision making, 19(1), 1-15.