Advances in Computer, Signals and Systems (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2026.100213
ISSN 2371-8838 Vol. 10 Num. 2

Computer Software and Information Technology Based on Big Data and Cloud Computing

Jiazheng Qiu

School of Computer and Software, Chengdu Neusoft University, Chengdu, Sichuan, China
3050822762@qq.com

Keywords: Computer Software and Information Technology; Big Data; Cloud Computing; Hadoop Distributed File System; HBase

Abstract: Traditional computer software and information technology research has faced many problems such as small data capacity, few computing resources, bad real-time performance and long development cycle. This study aimed at these problems and applied big data (BD) and cloud computing technologies to the processing of smart grid data. This study discussed the related key technologies such as Hadoop architecture, HDFS storage, SOA, HBase based unstructured data storage and cloud security mechanism based on message digest algorithm. The simulation experiment based on performance metrics compared the proposed method and traditional method. When the file size was 500 MB, the average CPU utilization was 33.8980%, the maximum original data detection result reached 499.11 MB, and the average time was 1191.6810 ms, which was better than the traditional approach. The experimental results show that the proposed method has better security, efficiency and timeliness. The research result shows the value of BD and cloud computing.

1. Introduction

Computer software and information technology are two main pillars of our modern day society. They are essential to business, medicine, energy production and consumption, and numerous other areas. As information technology advances, the amount of data collected by enterprises and organizations increases drastically. In the age of information, processing and analyzing big data is extremely important. Traditional computing architectures, however, face fundamental limitations in data scale, computational efficiency, real-time performance, and development cycle length, which restrict the broader application and advancement of computer software and information technology.

Previous research has proposed various solutions to address these challenges. Haji S. H. demonstrated that software defined networks provide greater flexibility, programmability, and centralized management compared to traditional networks, enabling better processing of large-scale complex data; however, real-time performance limitations remain significant [1]. Yoshida S. proposed a confidentiality parameterization method for information flow analysis in object-oriented programs to enhance data security, enabling the definition and monitoring of data flow during program design and improving access control strategies to curb potential data leakage risks [2]. Saifan Ahmad A. found that appropriately selecting feature subsets and adopting integrated

classification methods significantly enhances software defect prediction efficiency, though computational capacity for massive datasets still needs improvement [3].

The emergence of BD and cloud computing has brought revolutionary opportunities to the research of computer software and information technology. Wu C. studied time optimization of multiple knowledge transfers in BD environments, proposing a greedy algorithm-based scheme verified through numerical experiments to significantly reduce knowledge transfer time costs [4]. Andre J. C. explored convergence paths between BD and extreme-scale computing, aiming to develop shaping strategies for future software and data ecosystems for scientific inquiry, noting that their interaction brings enormous opportunities for cross-disciplinary cooperation [5]. Li J. proposed a secure attribute-based data sharing scheme using attribute encryption and proxy re-encryption to achieve secure data sharing and fine-grained control of access permissions in cloud environments, though processing power for large-scale datasets still requires improvement [6].

Currently, BD and cloud computing face several broad challenges. High-performance computing and storage architectures are required to support BD processing and analysis beyond the capacity of traditional systems [7]. Cloud computing applications must address issues including security, reliability, scalability, and cost-effectiveness [8-9]. These challenges affect both the technical development of cloud platforms and the business operations of enterprises and organizations. To address these multifaceted issues, this article proposes an integrated approach combining Hadoop-based BD data processing algorithms, HBase-based unstructured BD storage algorithms in cloud computing environments, and cloud security-based secure data storage. It is hoped that the analysis in this article will provide useful references and insights for the future development of computer software and information technology.

2. Computer Software Technology Based on BD

2.1 Types of BD Technologies in Computer Software

Big data (BD) refers to datasets whose volume and generation rate exceed the processing capacity of traditional data processing tools, requiring specialized distributed technologies for collection, storage, management, processing, and analysis [10-11]. BD processing technologies include data collection, data storage, data management, data processing, and data analysis, among which data analysis is the core component. At present, BD processing technology primarily adopts distributed computing frameworks such as Hadoop, SOA, Spark, and HDFS. The data collection, storage, processing, analysis, and application functions of BD technology are all achieved through computer software, enabling efficient processing and utilization of large-scale data .

(1) Hadoop file system architecture

Hadoop has been the most widely used BD technology framework. Its architecture consists of a master node responsible for managing file metadata and multiple slave nodes that store specific data blocks. This master-slave architecture provides excellent scalability for storing and processing massive amounts of data. Within the Hadoop architecture, large files are divided into several small data blocks dispersed across various slave nodes, and a redundant backup mechanism saves three separate copies of data across multiple slave nodes, ensuring data reliability and fault tolerance. These characteristics make Hadoop especially well-suited for distributed processing of large-scale smart grid datasets.

(2) SOA service architecture

SOA is a software architecture pattern that achieves business requirements by designing applications as sets of loosely coupled services communicating through standardized interfaces. The whole point of SOA is to generalize business functions into services that can be independently deployed. This makes systems more extensible and code more reusable. However, designing service

boundaries, service interface, versioning and security for SOA introduces challenges. Inter-service communications and data sharing should consider service security and reliability such as service availability and data confidentiality.

(3) HDFS storage

HDFS is a Hadoop-based distributed file system providing high reliability, high fault tolerance, high throughput, and low cost. Massive data is divided into several blocks stored across one or more nodes of a Hadoop cluster. When a client copies files, it communicates with the name node to transfer metadata and block information. The name node then provides space to certain data nodes which in turn receive data blocks from the client. The name node then returns confirmation of file upload to the client along with a persistence of metadata to disk. This ensures data persistence. Table 1 presents an example of HDFS file block information including file names, sizes, replication factors, and storage locations.

Table 1. HDFS file block information

File name	Size (MB)	Replication factor	Storage location
File 1	127	3	Node1, Node2, Node3
File 2	257	2	Node2, Node3
File 3	512	6	Node1, Node3, Node4
File 4	84	4	Node1, Node2, Node4

2.2 Multi-layer Network Architecture for BD Data Processing

Computer data processing algorithms based on BD adopt a multi-layer network structure comprising four layers: hardware device resources, resource virtualization, image and user management, and SOA service architecture. The complete multi-layer network architecture is illustrated in Figure 1.

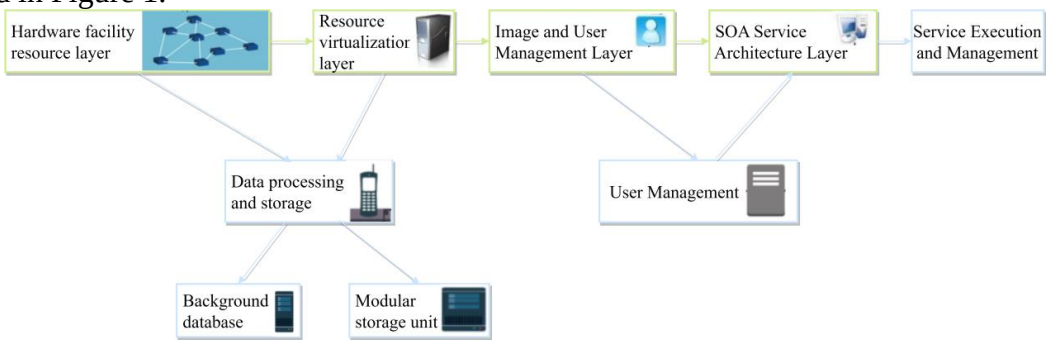

Figure 1. Multi-layer network architecture of BD computer data processing algorithms

The hardware device resource layer forms the lowest physical tier, composed of computers, databases, backend servers, and storage units connected through defined I/O interfaces, providing the foundation for virtual resource pool construction. The resource virtualization layer is established on backend servers to virtualize multiple computing hosts and data service hardware into data resource pools, network resource pools, and storage/computing resource pools. Upon receiving World Wide Web access, data processing, and storage requests from front-end users, the virtualization layer fully utilizes distributed virtual hardware resources to meet diverse user needs.

The image and user management layer establishes mappings between different data and manages user authorization and operation requirements hierarchically. User management includes user identification, request handling, and permission control. Image management covers image creation, deployment, library management, and lifecycle management, processing backend data and establishing mapping relationships between datasets. The SOA service architecture layer adopts a

plug-and-play arrangement of service components — including service registration, interfaces, workflow, access, and query — performing security checks on users before service provision, with clearly defined interfaces and business codes. This layered architecture ensures efficient, secure, and manageable BD processing across distributed environments.

2.3 Hadoop-Based BD Data Processing Algorithm

(1) Data collection

Smart grids are the new generation of power grids, which use information, communication and computer technology to efficiently and sustainably manage power. There are a large number of sensors and monitoring devices in smart grids that produce several gigabytes of data per second related to power quality, consumption, storage, transmission and distribution at different grid stations. Hadoop-based BD technology is particularly well-suited for this domain, enabling operators and managers to understand grid operation status and optimize grid performance through data mining and analysis. Taking a company's annual smart grid data as an example, data records are stored as triplets (measurement point marker, timestamp, numerical value), where values may be integers, floating-point numbers, strings, or byte arrays.

(2) Data processing

The data preprocessing module employs mean and smoothing methods to handle data with high errors and redundancy across wide area and local area networks, effectively filtering noise, repetitive, and null values. Based on the BD cloud service platform's virtualization resource pool, time coefficient α, load coefficient β, and cache coefficient γ are configured with preset values. Target values are assigned based on the weight of each data processing task in the virtual resource pool. Using the preprocessing formula by inserting many of the various BD processing tasks, a numerous amount of BD processing results are provided. If the computed values are less than or equal to the established threshold, normal data processing signals will be generated and output ensuring a high quality of data for later classification and storage.

(3) Virtual resource task scheduling

Smart grid data processing includes multiple links like huge data collection, analysis, processing, and storage; hence the requirement for substantial computing, storage, and network resources. Virtual technology helps the smart grids in managing and allocating these resources efficiently, improving the general efficiency of resource utilization. Task scheduling assigns specific virtual nodes to different tasks using distributed algorithms and grid algorithms to achieve data mapping and processing. Given y tasks assigned to z metadata nodes with single-task execution time t_j, execution time t_{ij} of the i-th task on the k-th resource is calculated based on resource capacity parameters. Total execution time T_{total} for all data processing tasks is minimized to ensure load balance and safe virtual resource operation throughout the processing pipeline.

(4) Data classification

This article employs a distributed computing-based classification hybrid method combining nearest neighbor classification algorithms, the MapReduce model, principal component analysis, and clustering algorithms to address smart grid BD classification[12-13]. Weighted Euclidean distances are calculated between n-dimensional data objects, with weights w_1, w_2, ..., w_k assigned according to the importance of each variable. Attribute normalization maps all attribute values to [0,1] to prevent large-number attributes from dominating the classification. The nearest neighbor metric determines whether two data objects are close neighbors. In the MapReduce model, the Map stage performs key-value mapping on classified datasets, calculating weighted Euclidean distances based on normalized attributes and classifying by similarity. The Reduce stage traverses all Map intermediate results and performs parallel classification. This divide-and-conquer approach

effectively enhances algorithm execution efficiency and reduces spatiotemporal complexity, with output results saved for secure cloud computing storage in the subsequent stage.

3. Computer Software Technology Based on Cloud Computing

Cloud computing provides computing resources, software, and data storage services through the internet, encompassing technologies such as automation, automatic scaling, and serverless computing [14-15].Cloud computing can give users with near instant access to computing resources, allowing them to quickly obtain the infrastructure they require, which also lowers the cost of constructing and maintaining enterprise IT facilities with improved levels of both efficiency and security in terms of resource utilization.

3.1 Unstructured BD Storage Algorithm Based on Cloud Computing

There are numerous types of unstructured big data, such as multimedia files (text, images, audio, video), emails, social media posts, and/or sensor data. Smart grid data, which is usually unstructured, includes sensor readings, equipment logs, monitoring data and alarm information. These data types are unpredictable and massive, requiring advanced processing and analysis techniques to extract valuable information for smart grid monitoring, scheduling, and optimization.

HBase uses HDFS as its underlying storage, supporting large-scale unstructured and semi-structured data through a data model similar to traditional relational databases. HBase divides tables into multiple regions, each containing a subset of rows with a primary node guiding installation and managing service regions. A master-slave distributed architecture enhances database scalability and ensures data consistency, improving application quality. In the unstructured BD storage algorithm, a binary classification evaluation standard is established using the cloud computing scheduling model. Assuming k BD sources with $k+\varepsilon$ nodes as carriers (ε is a constant greater than 0), each storage individual S_i ($i = 1, 2, ..., k+\varepsilon$) is represented as an independent row vector with values in [0,1], representing k different unstructured BD source resource packages. Through this formulation, cloud computing-based unstructured BD storage algorithms are developed and applied to efficiently manage the diverse data types in smart grid environments.

3.2 Data Security Storage Based on Cloud Security

In the cloud computing scenario, the data and applications are located in the service provider's data center and accessed by users through the Internet. The security issues of smart grid BD storage in cloud computing are focused on two aspects: Firstly, data confidentiality, integrity, trustworthiness and availability as well as user authentication, access control, security detection and emergency response should be addressed. Secondly, it is essential to offer secure guarantee for smart grid BD storage in the cloud.

The data encryption and storage process follows five steps: (1) Generate a digital digest — the message digest algorithm summarizes the data to be stored and generates a corresponding digital digest. (2) Encrypt the data — a key generation function obtains a random key, which encrypts the stored data to produce ciphertext. (3) Hide the random key — a password-based information hiding scheme conceals the random key. (4) Store the ciphertext — the encrypted ciphertext is uploaded to the cloud. (5) Save metadata — the key, digital digest, and filename are stored in HBase to complete the data storage process.

A random key hiding scheme is proposed using data source attributes and random padding values, with hash functions generating encryption keys. User passwords can be incorporated as attributes, allowing password updates by modifying only the random key rather than re-encrypting

stored data, thereby improving efficiency. Random padding also helps resist dictionary and pre-computation attacks. To ensure confidentiality, attribute combinations and hash details remain hidden. In HBase, the MetaTable stores RowKey, Timestamp, Metadata:Key, Metadata:Digest, and HiddenKey fields. A HiddenKey value of 0 indicates plaintext data, while 1 indicates encrypted data requiring decryption.

The data reading process includes: (1) retrieving ciphertext and metadata; (2) checking the HiddenKey field to determine whether decryption is needed; (3) restoring the random key using data source attributes; (4) decrypting the data; and (5) verifying integrity by comparing the generated ciphertext digest with the stored digest. Matching digests confirm data integrity, while inconsistencies indicate possible data tampering.

4. Experimental Results of Data Processing Methods Based on BD and Cloud Computing

4.1 CPU Usage Rate

CPU usage rate is an important index to reflect computing resource utilization. The higher CPU usage rate, the worse computational ability, and the lower data mining effect. So it is necessary to evaluate the computational ability of proposed method. Therefore, we select the data files with the size of 10MB, 50MB and 500MB to validate the CPU usage rate of our proposed method. As shown in smart grid data may cover different sizes. When the file size is 500MB, the average CPU usage rate of experimental group is 33.8980%, while the average CPU usage of control group is 39.2700%, namely the difference is about 5.4 percentage points. It can be seen that the computational burden of each node is better to be reduced by BD and cloud computing approach, and the whole computational ability is enhanced.

4.2 Data Security

To validate the effectiveness of the data security in the above method, we implemented dictionary attack and pre-computation attack on 100MB and 500MB data files, and the size of attack data and exception data were 20MB. When the file size was 500MB, the original data size of maximal original data detection result of experimental group was 499.11MB, and it was nearly close to original 500MB. And when the file size was 500MB, the original data size of maximal original data detection result of control group was 471.66MB. The 5.6% enhancement shows that the message digest and random key hiding scheme in the above method can resist the common attacks, and the data security is high, which offers stronger security guarantee for the sensitive smart grid operational data.

4.3 Processing Speed

Processing speed is the key to converting large data into useful information within a reasonable amount of time. For this reason, we measured the processing time of 10MB, 50MB, and 500MB files for the two groups. When the data size was 500MB, the average time taken by the experimental group to process the files was 1191.6810 ms and by the control group it was 1384.8100 ms i.e. the difference between the two groups was around 193 ms, which is 13.9% faster in terms of speed. This speed up has been achieved by MapReduce parallel processing model of Hadoop, which splits the large data set into smaller sub-tasks that can be performed in parallel on multiple nodes of the cluster, thus improving the speed of the data analysis workflow.

4.4 Parallel Processing Capability

Effective parallel processing is essential for large BD workloads. Parallel computing processes multiple tasks simultaneously, which results in a decreased total processing time and improved utilization rate of the employed resources. We measured the parallel processing time and memory usage rate of both groups. The maximum parallel processing time and maximum memory usage rate of the experimental group were 2.41 s and 29.98%, respectively, whereas those of the control group were 3.48 s and 33.47%. The results of parallel processing time reduction (30.7%) and memory usage rate (10.5%) show that the BD and cloud computing method divides data into multiple blocks and processes them in parallel, reduces the pressure of memory, avoids shortages of memory, and improves the stability and reliability of the whole system.

4.5 Accuracy

Accuracy denotes how accurate results are when using different data processing and analysis methods. inaccurate results may lead to wrong business decisions, and the enterprise impact could be huge. In table 2, we can show the precision and recall of two groups in six different iterations (from 10 to 60 iterations). Experimental group reaches to 0.89 as the highest recall and 0.95 as the highest precision while in control group 0.80 is the highest recall and 0.86 is the highest precision in all six iterations. The advantage of the performance shows the goodness of weighted Euclidean distance based classification hybrid algorithm and the goodness of distributed data preprocessing pipeline in filtering out the noise and redundancy from smart grid data.

Table 2. Precision and recall rates comparison between experimental group and control group

Iterations	Exp. Recall	Exp. Precision	Ctrl. Recall	Ctrl. Precision	Recall Δ	Prec. Δ
10	0.89	0.85	0.70	0.82	+0.19	+0.03
20	0.81	0.85	0.71	0.83	+0.10	+0.02
30	0.83	0.93	0.80	0.84	+0.03	+0.09
40	0.85	0.85	0.70	0.85	+0.15	0.00
50	0.85	0.86	0.75	0.83	+0.10	+0.03
60	0.81	0.95	0.74	0.86	+0.07	+0.09

5. Discussion

Experimental results for these four evaluation indicators show that the proposed BD and cloud computing approach achieves better performance than the comparison approaches. Specifically, the experimental results show that the CPU usage rate is reduced from 39.27% at 500MB to 33.90% after virtual resource pooling and task scheduling spreads the computation task to 23 Hadoop nodes to avoid a single server bottleneck, the processing time is reduced by 13.9% because when MapReduce parallel processing significantly speeds up the real-time smart grid monitoring data analysis from large volumes of data, parallel processing results show that the processing time is reduced by 30.7% and 10.5% memory is used for saving more resources from.

From the security perspective, the storage framework of the proposed cloud-based storage scheme combining message digest algorithms and random key hiding can resist dictionary and pre-computation attacks and preserve almost perfect data integrity (499.11MB detected from 500MB files), the high accuracy of the distributed algorithm satisfies the requirements of smart grid analytics because when precision could be up to 0.95 and recall could be up to 0.89.

6. Conclusions

With the rapid development of big data and cloud computing, enterprises begin to use these two technologies to collect, transfer and process data, and this paper studied Hadoop based BD processing, HBase based unstructured data storage in cloud environment and secure cloud data storage using smart grid data as an experimental example. The simulation experiment based on Hadoop platform of 23 nodes compared the proposed method with traditional method, the experimental results showed that CPU usage was lower (33.90% versus 39.27%), original data can be detected better (499.11 MB versus 471.66 MB), the processing time was shorter (1191.68 ms versus 1384.81 ms), parallel processing time was shorter (2.41 s versus 3.48 s), memory usage was lower (29.98% versus 33.47%), and classification performance achieved precision of 0.95 and recall of 0.89, so the experimental results demonstrated that the integrated BD and cloud computing method can achieve better accuracy, efficiency and security than traditional method, and the related future work should focus on scalability, real-time processing and cloud security for more larger and more complex data.

References

[1] Haji S. H., Zeebaree S. R. M., Saeed R. H., Ameen S. Y., Yasin H. M.. "Comparison of Software Defined Networking with Traditional Networking." Asian Journal of Computer Science and Information Technology 9.2(2021):1-18.

[2] Yoshida S., H. Kuwabara, and Y. Kunieda. "Secrecy Parameterization in Information Flow Analysis for Object-oriented Programs." Computer Software 36.1(2019):48-65.

[3] Saifan Ahmad A., L. A. Abuwardih. "Software Defect Prediction Based on Feature Subset Selection and Ensemble Classification." ECTI Transactions on Computer and Information Technology (ECTI-CIT) 14.2(2020):213-228.

[4] Wu C., Zapevalova E., Chen Y., Li F. "Time Optimization of Multiple Knowledge Transfers in the Big Data Environment." Computers Materials & Continua 54.3(2018):269-285.

[5] Andre J. C., Antoniu G., Asch M., Sala R. B., Zacharov I.. "Big Data and Extreme-Scale Computing: Pathways to Convergence - Toward a Shaping Strategy for a Future Software and Data Ecosystem for Scientific Inquiry." International Journal of High Performance Computing Applications 32.4(2018):435-479.

[6] Li J., Zhang Y., Chen X., Xiang Y. "Secure attribute-based data sharing for resource-limited users in cloud computing." Computers & Security 72.JAN.(2018):1-12.

[7] Xu, W., Zhou, H., Cheng, N., Lyu, F., Shi, W., Chen, J., et al. "Internet of Vehicles in Big Data Era." IEEE/CAA Journal of Automatica Sinica 5.1(2018):19-35.

[8] Wei, W., Fan, X., Song, H., Fan, X., Yang, J. "Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud Computing." IEEE Transactions on Services Computing 11.99(2018):78-89.

[9] Namasudra, Suyel. "An improved attribute-based encryption technique towards the data security in cloud computing." Concurrency and Computation: Practice and Experience 31.3(2019):4364.

[10] Dai, H. N., Wang, H., Xu, G., Wan, J., & Imran, M. "Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies." Enterprise Information Systems 14.9-10 (2020): 1279-1303.

[11] Gao, R. X., Wang, L., Helu, M., & Teti, R. "Big data analytics for smart factories of the future." CIRP Annals - Manufacturing Technology 69.2(2020):668-692.

[12] Madan, Suman, P. Goswami. "k-DDD Measure and MapReduce Based Anonymity Model for Secured Privacy-Preserving Big Data Publishing." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 27.2(2019):177-199.

[13] Huawei, Y. I., Zaiseng, N., Fuzhi, Z., Xiaohui, L. I., Yajun, W. "Robust Recommendation Algorithm Based on Kernel Principal Component Analysis and Fuzzy C-means Clustering." Wuhan University Journal of Natural Sciences 23.002(2018):111-119.

[14] Ainapure, Bharati, D. Shah, A. A. Rao. "Adaptive Multilevel Fuzzy-based Authentication Framework to Mitigate Cache Side Channel Attack in Cloud Computing." International Journal of Modeling, Simulation, and Scientific Computing 9.5(2018):1850045.1-1850045.21.

[15] Abdel-Basset, Mohamed, M. Mohamed, V. Chang. "NMCDA: A framework for evaluating cloud computing services." FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE 86.SEP.(2018):12-29.