

Design of Storage Algorithm for Energy Industry Internet Information Cluster Under Edge Device Access

Qing LI*, Peng YU, Anzi HUANG, Haifeng QIU, Mengqiu YAN

Shenzhen Power Supply Bureau Co., Ltd., 39 Zhongxin 1st Road, Futian District, Shenzhen 518100, China
liq518100@163.com (*Corresponding author), yupeng019@163.com, huanganzi96@163.com,
qiuhaifeng02@163.com, yanmengqiu01@163.com

Abstract: This paper proposes a cluster storage algorithm for energy industry Internet information that would enable the access of edge devices. This process includes, among others, the creation of a storage framework for the energy industry Internet information, and collecting various data by sensors, and also performing edge fusion and calculations on the data obtained by the sensor through the edge equipment. Through the UDP-based Data Transfer (UDT) protocol of the data transfer layer, the data is stored in the server cluster by using the object-based cluster storage file system CFSlight. The dynamic load balancing strategy is adopted in order to transfer the energy industry Internet data on the nodes with a heavy load to the nodes with a light load, so as to achieve the load balancing in the context of the server cluster data storage. The experimental results show that the data after edge device fusion and after cluster storage can be read smoothly, and it is accurate, without any loss and pollution. The proposed algorithm can achieve cluster storage for the Internet information in the energy industry, with a short storage time and a good load balancing effect.

Keywords: Energy industry Internet, Edge equipment, Edge calculation, Data transmission, Cluster storage, Load levelling.

1. Introduction

Energy industry Internet refers to a new energy system that provides digitalization, intelligence and the interconnection of energy production, transmission, supply and consumption through information technology and Internet technology (Bi et al., 2025; Prakash et al., 2022; Orowski, Cofta, & Orlowski, 2022). Due to the rapid development of energy industry Internet, all the generated data has exploded in geometric series, and all online data has grown rapidly. Storing massive energy industry Internet data has become difficult, and at the same time, harder requirements have been put forward with regard to the performance of storage systems (I/O bandwidth, throughput, response time, etc.) (Kaliyamoorthy & Canessane, 2021). Data storage refers to the permanent preservation and management of data, including not only the database, but also a variety of simple storage data types, such as simple files (Ovesen et al., 2021), e-mails (Fabian, et al., 2020) etc. and it is composed of a series of bytes managed by a database management system (DBMS) or a file system (Alves, Mateus-Coelho & Cruz-Cunha, 2021; Zhang et al., 2021). Therefore, both databases and files can be regarded as a series of bytes (Achour, Benattou & Lanet, 2021). In order to perform the efficient operation and large-scale data processing of energy industry Internet, it is very important to develop an information cluster storage technology.

Further on, many other researchers have focused on the field of data storage. Dharminder et al.

(2022) proposed a post-quantum integer lattice-based signature encryption storage scheme for the Internet of Things. Vishwakarma et al. (2022) studied cloud data storage based on an improved resource scheduling of the security system. First, they solved the complex optimization problem by employing the reptile search algorithm (RSA) with a gradient-free method, and improved the resource scheduling program of cloud storage, which enabled the cloud to store data more quickly. Mass data storage for IOT devices based on blockchain technology was studied by Maftai et al. (2023), which builds a distributed data storage architecture through blockchain technology, eliminates the need for centralized network topology, utilizes the advantages of blockchain and improves the efficiency of data storage through automation, and its scalability can adapt to the storage of a large amount of data. Ye & Park (2021) studied reliable data storage based on blockchain and IPFS. The data can be well protected and stored by accessing the storage system and managing the data through the distributed application on Ethereum. McManamon et al. (2021) studied data storage under the conditions of high-capacity storage and high access rate, and used optical communication to store data. As the speed of light is the highest, it is reasonable to use this method to store pb or even eb data, and its format is more compact than that of traditional data storage means. However, even if the above methods can complete data storage, there are still certain shortcomings, such as a slow data storage, possible data loss or damage during

storage, the inability to modify data, and the risk of data leakage.

The data stored in a cluster features several advantages, such as an easy management, a fast storage speed, data storage in different servers to reduce the risk of data leakage, a strong expansion ability and a simple establishment of server clusters. Edge Computing allows devices to complete data collection and preprocessing in the local network by deploying computing nodes at the edge of the local network, thus overcoming the problems of low efficiency and large transmission delay for cloud computing for massive primary data. Therefore, this paper proposes an energy industry Internet information cluster storage algorithm with connected edge devices, and uses edge devices to fuse the data of different sensors set in the energy industry. The UDP-based Data Transfer (UDT) Protocol is used for transferring the data of edge devices to the service cluster to complete the data storage. When data is stored in the server cluster, the load balancing algorithm is employed to ensure that the servers in the cluster will be loaded in a balanced manner.

The remainder of this article is organized as follows: Section 2 details the cluster storage algorithm, constructs a four-layer framework including data collection, edge fusion, data transmission, and cluster storage, it also explores edge fusion and the computing architecture, and explains the UDP-based Data Transfer (UDT) Protocol format, transmission and reception process, congestion control, as well as the CFSlight file system and the dynamic load balancing strategy. Section 3, which is dedicated to experimental verification, compares data fusion, the transmission efficiency, storage time, and load balancing performance based on real datasets and the OMNeT++ environment. Finally, Section 4 summarizes the contributions and highlights the advantages of algorithms in storage efficiency and system stability, providing practical solutions for large-scale energy data management.

2. Energy Industry Internet Information Cluster Storage

2.1 Information Cluster Storage Framework Construction

The energy industry processes a large amount of key data, such as data on energy production,

transmission and consumption. As such, by proposing an information cluster storage framework, this paper aims to ensure the consistency and reliability of data, to avoid data loss and damage, to improve the efficiency of data storage and data processing, and to provide high-reliability data storage. This framework consists of four layers, namely: (1) Data acquisition layer: using sensors to collect various data from the energy industry Internet (Deji, Choudhary & Sharma, 2022); (2) Edge computing layer: edge calculation and data fusion for the data obtained by the sensor in the edge device; (3) Data transfer layer: data transmission is carried out by using the UDP-based data transmission protocol (Sumathi & Anitha, 2020) and (4) Cluster storage layer: the cluster file system CFSlight based on object storage is used for energy industry Internet information cluster storage.

The energy industry Internet information cluster storage framework described above is shown in Figure 1.

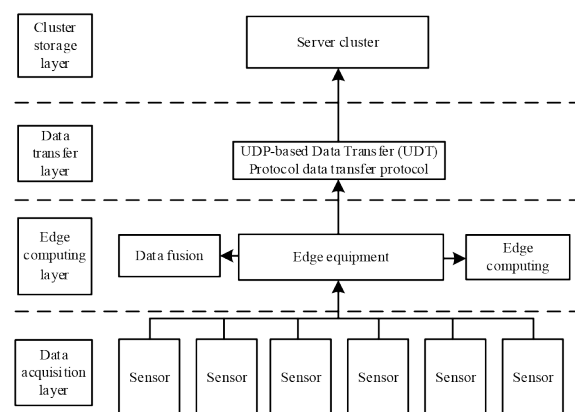


Figure 1. Information cluster storage framework

2.2 Edge Computing Layer and Information Fusion for Energy Industry Internet

The data related to the energy industry features high confidentiality and security requirements. By computing on edge devices, sensitive data can be kept locally or in the near field for processing, thereby reducing the risks and potential security vulnerabilities in the transmission of sensitive data. The energy industry Internet involves many devices and sensors, and generates a large amount of data. However, this data may include redundant information, for example, in the case when multiple sensors measure the same data.

2.2.1 Data Edge Fusion

Based on the above-mentioned information cluster storage framework, sensors in the data acquisition layer are used for collecting a variety of data on the energy industry Internet. However, because the collected data usually features a high frequency and a large number of properties, if this data is directly transmitted to the cloud for processing, it will lead to network congestion and delay increase. By fusing data at edge nodes, the the frequency of data transmission in the cloud can be reduced, the network load can be reduced, and the data processing efficiency can be improved. Therefore, it is assumed that the data acquisition layer is provided with n different types of sensors (Sain et al., 2021) which are recorded as data sets $\{D_1, D_2, \dots, D_n\}$. A comprehensive analysis of the factors affecting the stable operation of the energy industry Internet denoted by ρ , was carried out. To measure various parameters affecting the operation of the energy industry Internet, many types of sensors are often set to measure different parameters from different angles (Sun & Wan, 2022), such as a temperature sensor for measuring temperature, a pressure sensor for measuring pressure, a vibration sensor for measuring vibration, an angle sensor for measuring different angles, a current sensor for measuring current etc. Based on this information, the state of the energy industry Internet can be assessed from different angles.

Due to the changeable environment of the energy industry Internet, it is easy to interfere with the normal operation of the energy industry Internet. Therefore, various problems can be measured based on the probability theory. It is supposed that in the timespan t , the sensor set $\{D_1, D_2, \dots, D_n\}$ measured parameters that affect the normal operation of the energy industry Internet and which are recorded as data sets $\{\rho_1(t), \rho_2(t), \dots, \rho_i(t), \dots, \rho_n(t)\}$, where i denotes the i -th sensor. In order to improve the measurement accuracy, the parameters that affect the normal operation of the energy industry Internet are weighted. $\bar{\rho}$ represents the results of the weighting and the weighting formula is as follows:

$$\bar{\rho} = \sum_{i=1}^n \omega_i \rho_i = [\omega_1, \omega_2, \dots, \omega_n] [\rho_1, \rho_2, \dots, \rho_n]^T \quad (1)$$

Here, because each sensor has a weighted value, the weighted value of the i -th sensor

can be expressed by ω_i . If ρ_i is set as a random variable in the calculation process, it obeys normal distribution in the probability theory $N(\mu_i, \sigma_i^2)$. From the perspective of the multivariate probability distribution theory, the data sets $[\rho_1, \rho_2, \dots, \rho_n]$ obey the metanormal distribution $N(a, B)$, where the expected value of the normal distribution is denoted by $a = [\mu_1, \mu_2, \dots, \mu_n]$ and $B = \text{diag}[\sigma_1, \sigma_2, \dots, \sigma_n]$ is the covariance matrix.

The distribution density function provides a description of the statistical characteristics of parameter distribution, such as mean, covariance, etc. By analyzing these statistical characteristics, the efficiency of collecting operational and maintenance data for the energy industry Internet can be optimized. Therefore, for data sets $[\rho_1, \rho_2, \dots, \rho_n]$ a linear combination should be performed to obtain information about the distribution density function $\bar{\rho}$, so as to accomplish the edge fusion of the collected data, which can be expressed by the following formula:

$$p(y) = \frac{\bar{\rho}}{(2\pi)^{\frac{n}{2}} \left(\sum_{i=1}^n \omega_i^2 \sigma_i^2 \right)^{\frac{1}{2}}} \exp \left\{ - \frac{\left(\sum_{i=1}^n \omega_i \mu_i \right)^2}{\sum_{i=1}^n \omega_i^2 \sigma_i^2} \right\} \quad (2)$$

where $\bar{\rho}$ is considered a random variable after sensor detection data fusion, when it conforms to the normal distribution $N\left(\sum_{i=1}^n \omega_i \mu_i, \sum_{i=1}^n \omega_i^2 \sigma_i^2\right)$, μ_i denotes the mathematical expectation value of probability distribution, through which the calculation errors of different sensors can be comprehensively reflected. After data fusion, the expected value of the distribution of the data detected by sensors can be expressed as:

$$\bar{\mu} = \sum_{i=1}^n \omega_i \mu_i \quad (3)$$

The standard deviation can be recorded as:

$$\bar{\sigma} = \sqrt{\sum_{i=1}^n \omega_i^2 \sigma_i^2} \quad (4)$$

2.2.2 Edge Calculation Model

After the edge fusion of various types of sensor data, in order to improve the data processing and applicability of the energy industry Internet, an edge computing model is connected to the edge of the network, and a large amount of energy

industry Internet data is calculated and analyzed, so that the calculation result can be closest to the data collection point, which is convenient for the real-time control of the massive data in the operational process of the energy industry Internet and improves the operability of the energy industry Internet.

Usually, the energy industry Internet needs to respond to real-time events and data changes in a timely manner. Using sensors to collect data, perform edge fusion, and calculate the delay of the edge model can push real-time data processing and analysis to the network edge, reduce the delay of data transmission to the cloud, and enable faster real-time operational decision-making, such as fault warning, equipment condition assessment, and energy scheduling, so as to meet real-time requirements. The architecture of the edge computing equipment features an advanced DVFS (Dynamic Frequency and Voltage Scaling) technology (Tsumachi et al., 2021), and when using this algorithm for calculations, it adopts edge computing as a service mode (Chen et al., 2025), which can adjust energy consumption by changing the cycle frequency of the CPU. In the concrete calculation process, it is assumed that f_m denotes energy consumption in the process of edge calculation, where f_m is restricted by the maximum computing power of the edge device, denoted by $f_{CPU}^{\max}$. When the delay needs to be reduced, f_m must be increased to ensure the real-time performance of edge tasks, otherwise, f_m can be reduced. It is supposed that there is a task checking model $A(L, T_d, X)$, where L denotes the data packet size for the energy industry Internet data fusion, and its unit is bit, T_d is the energy industry Internet data edge calculation time, expressed in seconds, and X represents the workload of edge calculation. The delay of the edge calculation model is determined by the following formula:

$$t_m = \frac{LX}{f_m \cdot p(y)} \quad (5)$$

In this formula, $f_m \leq f_{CPU}^{\max}$.

The relationship between the calculation performance and time delay t_m can be obtained by using equation (5). Using the relationship between edge calculation time and energy consumption introduced above, the edge model can be expressed as:

$$E_s = \max_{1 \leq i \leq n} \sum_{i=1}^m f_i \cdot t_{total} \quad (6)$$

2.3 Energy Industry Internet Information Transmission Through the UDP-based Data Transfer (UDT) Protocol

After the edge calculation and data fusion are completed in the edge device, this data is sent to the server cluster in the cluster storage layer for storage through the data transfer layer based on the transmission protocol model known as the UDP-based Data Transfer (UDT) Protocol. For the energy industry Internet, the network conditions may change, leading to Internet packet loss, delay increase or bandwidth fluctuation. The UDP-based Data Transfer (UDT) Protocol has some adaptive mechanisms, which can adjust and optimize the transmission strategy according to the network conditions to ensure the reliability of data and the efficiency of transmission.

The UDP-based Data Transfer (UDT) Protocol, a protocol model suitable for large-capacity data transmission, is divided into four layers: (1) UDP Socket layer; (2) Cross-platform Socket interface layer; (3) UDP-based Data Transfer (UDT) Protocol interface layer; (4) Application layer.

The UDP Socket layer at the bottom is responsible for the UDP data communication of the basic energy industry Internet. It receives any UDP data packet from the energy industry Internet and passes it to the upper layer. At the same time, it also replaces the upper layer to send UDP packets of the energy industry Internet. The cross-platform Socket interface layer encapsulates the Socket API, which works under different operating system platforms, providing a unified access interface. The UDP Interface layer provides an access interface for the UDP-based Data Transfer (UDT) Protocol to the application program in an object-oriented way. Through this layer, the application program can conveniently call the the UDP-based Data Transfer (UDT) Protocol function, establish a UDP-based Data Transfer (UDT) Protocol connection, and send or receive energy industry Internet data. The structure of UDP-based Data Transfer (UDT) Protocol is shown in Figure 2.

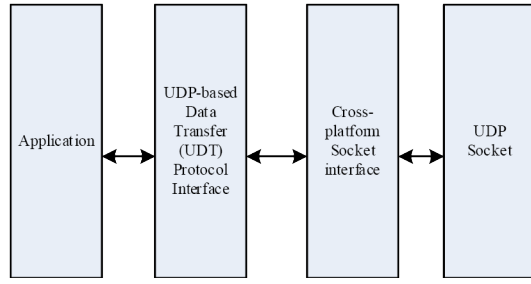


Figure 2. UDP-based Data Transfer (UDT) Protocol structure

2.3.1 Energy Industry Internet Data Packet Form

After data information fusion, in order to ensure the integrity and reliability of data in the transmission process, the energy industry Internet data is packaged into data packets. A data packet has a certain structure, including the data header, data body and checksum field, which can be used to verify the integrity of data, detect and correct errors in transmission, help prevent data loss, damage or tampering, and improve the reliability of data transmission. There are two kinds of UDP-based Data Transfer (UDT) Protocol packets in energy industry Internet: DataPacket and Control Packet. They all have a common packet header as shown in Figure 3.

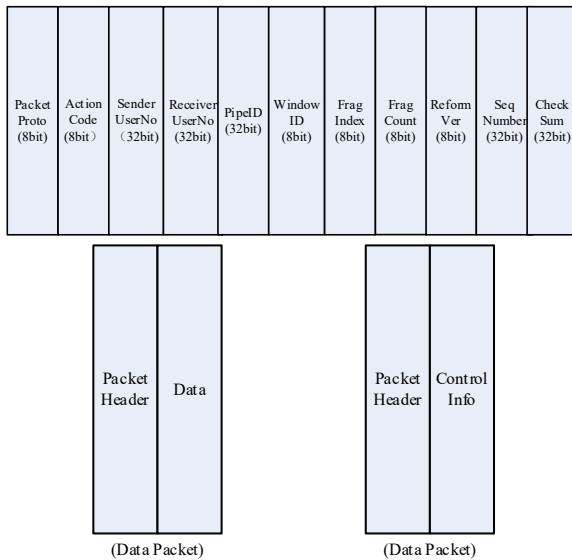


Figure 3. Packet format

The packet header fields for the energy industry Internet data packets shown in Figure 3 are defined as follows:

PacketProto: protocol type;

ActionCode: action code;

SenderUserNo: the user number of the sender;

ReceiverUserNo: the user number of the receiver;
PipeID: pipeline ID (the receiving pipeline ID is unique locally);

WindowID: window ID (increasing in order after connection) ;

FragIndex: the sequence number of the segment in the window (0-based);

FragCount: the total number of segments in the window;

ReformVer: the internal reorganization version number of the window (1-based);

SeqNumber: the sequence number of packets, which is usually incremented by 1 for each packet;

Checksum: checksum.

2.3.2 Energy Industry Internet Data Receiving and Sending Process

The receiving process is as follows:

1. The “receiver thread” detects the “receiving pipeline” and tries to read the energy industry Internet data from it. If no data has arrived, it waits.
2. The system receives the UDP-based Data Transfer (UDT) Protocol data packet of the energy industry Internet, and puts the energy industry Internet data packet into the corresponding receiving buffer according to the contents of the energy industry Internet data packet.
3. The receiving buffer combines the energy industry Internet data packets into a “data window”.
4. If the successfully created data window can just be connected to the previous data window, it will be placed in the Available List, otherwise it will be placed in the Sorted List.
5. The “receiving pipeline” reads the energy industry Internet data window from the “available list” and hands it over to the receiver.

The sending process is as follows

1. The Internet data of the energy industry to be sent is placed into the “sending pipeline”.
2. The “sending pipeline” detects the status of the “sending buffer” in real time, and when there is spare space in the buffer, it transfers the internet data of the energy industry to be sent to the “sending buffer”. Otherwise, it waits.

3. The “sending buffer” divides the data to be sent into several “data windows”.
4. The “sender thread” scans the sending buffer and handles the following tasks:
 - a. sending each data window;
 - b. processing data packet confirmation and overtime retransmission;
 - c. counting the current packet loss rate;
 - d. controlling the “sending governor” to adjust the current sending rate in real time to avoid network congestion.
5. If the data confirmation of an energy industry Internet data window is received, the data window is deleted from the sending buffer.

2.3.3 Internet Data Transmission Congestion Control in Energy Industry

UDP-based Data Transfer (UDT) Protocol uses the unreliable UDP protocol to send data from the underlying energy industry Internet, so UDP-based Data Transfer (UDT) Protocol must deal with issues such as data packet retransmission and timeout detection. In order to ensure transmission efficiency on the premise of saving bandwidth, BUTP must control the data transmission rate of energy industry Internet in real time to prevent congestion in the UDP-based Data Transfer (UDT) Protocol. Congestion Control is abbreviated as CC. The congestion control for Internet data transmission in energy industry is shown in Figure 4.

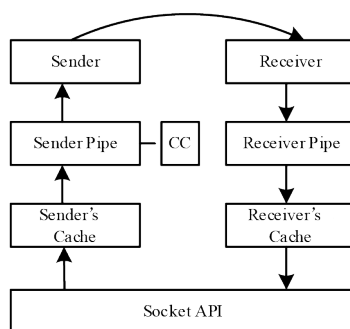


Figure 4. Network congestion control

Under the given network conditions, after the UDP transmission rate reaches a certain value, continuing to increase the transmission rate will lead to a higher packet loss rate. So a high transmission rate does not necessarily mean a high transmission efficiency. In the UDP-based Data

Transfer (UDT) Protocol, the sender is responsible for congestion control. The sender has a “sending governor”, which monitors the current energy industry Internet data transmission rate and packet loss rate in real time, reduces the energy industry Internet data transmission rate when the packet loss rate exceeds a certain value, or gradually increases the energy industry Internet data transmission rate. In the actual energy industry Internet data transmission process, the transmission rate will be roughly stable within a certain range after being controlled by the governor, so as to make full use of the current bandwidth and avoid unnecessary network congestion.

2.4 Energy Industry Internet Information Cluster Storage Based on CFSlight

After receiving the energy industry Internet data transmitted through the UDP-based Data Transfer (UDT) Protocol protocol, the cluster storage layer uses the CFSlight method to store the data. The CFSlight method has an efficient storage management ability, which is optimized for the characteristics of Internet data in energy industry and performs high-speed data writing and reading operations. By using the CFSlight method, fast data storage and retrieval can be accomplished, and the storage performance and response speed of the system can be improved. The cluster storage method CFSlight includes three parts, namely, the client file system, the metadata control system and object storage system. The client file system is the interface of the CFSlight file system, the metadata control system manages the attribute directory information and file description information of the CFSlight file system, and the object storage system is responsible for the storage of data files.

2.4.1 Client File System

The Client file system can store data on a local disk, and enable a fast and efficient data access. Compared with storing data in a remote server or in a cloud, using a local client file system can reduce network transmission delay and bandwidth consumption, and improve the data reading and writing speed and system response performance. The client file system of the cluster file system can be understood as an extensible file system. It can be inserted in the VFS (virtual file system) of Linux to provide an interface for

the users of the file system. The focus of the client file system is on improving the consistency of the global namespace. The CFSlight client file system obtains a logically unified cluster file system space by accessing the metadata control system and the object storage system, as shown in Figure 5.

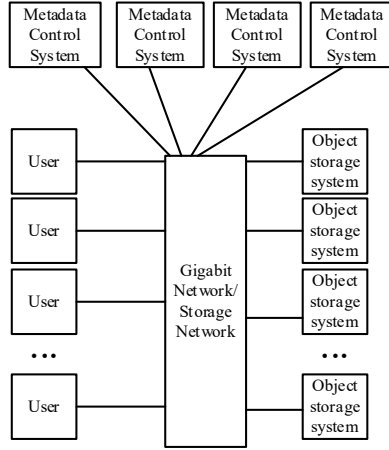


Figure 5. Cluster storage structure

2.4.2 Metadata Control System

A cluster is formed by a plurality of metadata servers (MDSs), and the directory for storing energy industry Internet data files is obtained, including the root directory of the energy industry Internet data cluster file system. The energy industry Internet data file objects distributed on different object storage bodies are organised into a tree-structured and unified storage space. In contrast to the traditional Unix file system, the directory of CFSlight is no longer stored in file objects, but stored in metadata servers independently, so as to facilitate the centralized management of the directory and access control at the directory level.

2.4.3 Object Storage System

Many works on the reliability and fault tolerance of the existing large-scale storage systems are based on the RAID technology, such as OceanStore, which guarantees the reliability and load balance of Internet data in energy industry through erasure coding and replication. In the CFSlight system, the existing Linux file system is used for managing the defined object entities, which can easily increase the scalability of the system. The reliability and fault-tolerance design of the object storage system will be addressed in a future work.

2.4.4 Relationship Between Subsystems

According to the object-oriented design concept, the three main components of this system feature data interaction, and cooperate to complete different tasks for maintaining the energy industry Internet data file system and enabling file access. As an example, the relationship between the client file system and the metadata control system is based on metadata exchange. All the operations related to the energy industry Internet metadata of the client file system can only be performed by the metadata control system. The data interaction between the components of the system is shown in Figure 6.

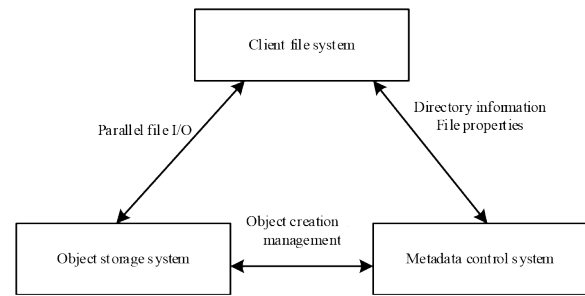


Figure 6. Subsystem relationships

2.5 Server Cluster Load Balancing

In order to prevent the load imbalance between the MDS nodes of the metadata server in the server cluster of the cluster storage layer and enhance the efficiency of Internet information storage in energy industry, it is necessary to adjust the load of the server cluster to make it balanced. By assessing the difference between the request response time of each MDS in the MDS cluster and the average response time of the whole MDS cluster, it is assessed whether the MDS load is balanced. When the load between different MDSs in the MDS cluster is unbalanced, dynamic load balancing should be adopted in order to copy or transfer some file metadata on the MDS with a heavy load to the MDS with a light load, so as to achieve the load balancing of the server cluster.

2.5.1 MDS Request Response Time

Let $R(x)$ denote the response time of MDS_x to an energy industry Internet metadata request for a file, where $x \in M(n)$. Generally speaking, each MDS can be modeled by the M/G/I queuing model. Because the energy industry Internet metadata of each file in MDS is structured and the data size is equal, and assuming that each MDS

has the same processing capacity, the service time for each file's energy industry Internet metadata request is the same, and based on the assumptions above, the response time of MDS_x to an energy industry Internet metadata request for a file can be expressed as:

$$R(x) = \frac{\lambda_x \cdot B_x}{E_s \cdot (1 - \tau_x)} + T_x \cdot E_s / D(R) \quad (7)$$

Here, T_x is for MDS_x the service time for the energy industry Internet metadata request, λ_x is for MDS_x the arrival rate of Internet metadata requests in the energy industry, τ_x is for MDS_x the Internet metadata access intensity for the energy industry and B_x is for MDS_x the Internet metadata processing ability of the energy industry.

Here, the average access intensity of the MDS cluster $\tau(R)$, the request service time $T(R)$ and the response time variability $D(R)$ can be expressed as:

$$\tau(R) = \frac{1}{N} \sum_{i=1}^N \tau_i \quad (8)$$

$$T(R) = \frac{1}{N} \sum_{i=1}^N R_i \quad (9)$$

$$D(R) = \frac{1}{N} \sum_{i=1}^N T(R)^2 \quad (10)$$

2.5.2 MDS Dynamic Load Balancing Implementation

According to the calculation results for the response time related to the energy industry Internet metadata request, in the object storage system, the load balance of energy industry Internet metadata between MDSs is completely distributed, and MDS nodes need to exchange their mutual load information periodically. The goal of MDS dynamic load balancing is to make the response times of each MDS to the file metadata request not much different. This paper uses the following function to measure the load imbalance of MDS_x :

$$M(x) = R(x) - T(R) \quad (x \in M(n)) \quad (11)$$

where the function $M(x)$ refers to the difference between MDS_x 's request response time and the average response time of the MDS cluster which reflects the degree of load balance of MDS_x .

A load difference threshold M_{ALARM} is defined in the MDS cluster, if $\exists \forall x \in M(n)$ so that

$M(x) \geq M_{ALARM}$, the dynamic load balancing algorithm will be applied to MDS.

The algorithm process is as follows:

When $M(i) \geq M_{ALARM}$, $i \in (1, n)$, obtaining a low-load MDS server L_MDS should make L_MDS satisfy $\left| M(i) + \sum_{j \in M_MDS} M(j) \right| < \left| M(i) + \sum_{k \in SL_MDS} M(k) \right|$, $\forall SL_MDS \subseteq L_MDS$, where M_MDS is the moderate load, and then part of the MDS_i in file metadata is transferred to MDS_j . The total request rate for the transferred file metadata between MDS_i and MDS_j can be expressed as:

$$\frac{M(x)}{(2 \cdot M(x) - M_{ALARM}) \cdot M_{ALARM}} \quad (12)$$

Based on the above process, it can be stated that dynamic load balancing is obtained when the threshold M_{ALARM} exceeds a certain value. This can reduce the frequent migration of energy industry Internet metadata in files between MDSs, because dynamic load balancing is activated only when necessary. After dynamic load balancing is started from the most overloaded MDS_i , the energy industry Internet metadata of the files with medium overload is allocated to the appropriate MDS with light load according to a certain proportion, which finally makes the MDS in the system load balanced.

3. Experimental Analysis

3.1 Experimental Subjects

Taking a city's energy industry Internet as the experimental object, the coverage of the energy industry Internet includes four county-level power supply companies, with 42 central power supply offices and 55 power supply business halls, serving more than 2.28 million power customers. There are 112 substations of 35kV and above under the jurisdiction of the selected city, with a substation capacity of 12,665,400kVA and a line length of 3,535.6261 km among them, 20 substations of 220kV with a capacity of 6.78kV and a line length of 1,154.484km. 47 substations of 110kV with a transformation capacity of 4.8 million KVA and a line length of 1,269.7025 km and 45 substations of 35kV with a capacity of 1,085,400 kVA and a line length of 1,149.8242 km. To that, there are 18,328 public transformers of 10 kV with a capacity of 7,895,450 KVA and a line length of 14,372 km. In the energy industry Internet, it involves a variety

of edge devices and information cluster storage components, and OMNeT++ can meet the needs of system scalability modeling and simulation. The information generated by this method is stored in clusters, and the application effect of this method is analyzed from the perspectives of data collection, transmission and storage.

3.2 Experimental Data

For convenience purposes, this dataset is referred to as the “operation dataset of the energy industry Internet in a certain city” in this paper and serves as the basic data source for the experiment. To verify the effect of the edge computing data fusion method proposed in this paper, sensor data from 10 randomly selected power stations of the energy industry Internet was collected in July 2024. The data was fused using the method proposed in this paper, and the results are shown in Table 1.

As it can be seen from Table 1, the edge device sorted out the data from 10 power station sensors in a certain area of the chosen city in July 2024, and parameter types (i.e. temperature, pressure, thermal power generation, and photovoltaic power generation), thereby structuring the data for its subsequent storage and retrieval.

In order to verify the efficiency of the data transmission protocol applied in this paper, taking time as the experimental index, the transmission speed of the proposed UDP-based Data Transfer (UDT) Protocol was compared with that of the TCP/IP, HTTP and SMTP protocols, which proved that the lower the transmission time, the higher the data transmission efficiency of the energy industry Internet, as shown in Figure 7.

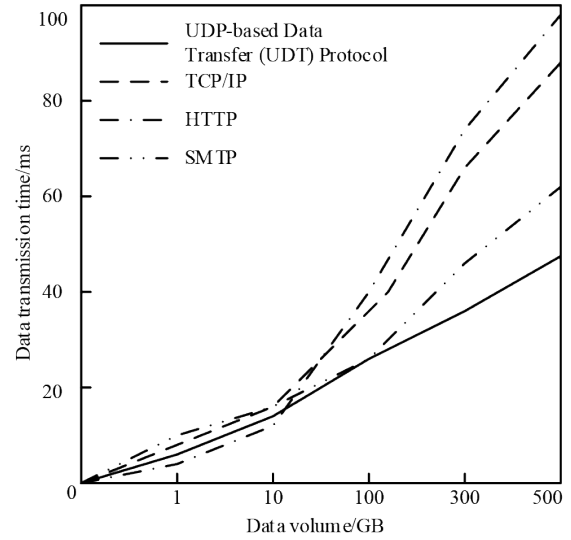


Figure 7. Data transmission speed comparison

As it can be seen from Figure 7, the transmission speed of the HTTP data transmission protocol is the highest when the data volume is lower than 10GB, but when the data volume continues to increase, the data transmission time of the HTTP data transmission protocol greatly increases. When the data volume reaches 500GB, the data transmission time of the HTTP data transmission protocol is about 100ms, and the data transmission time of the TCP/IP data transmission protocol is about 90ms. When the amount of data is small, the data transmission time of the UDP-based Data Transfer (UDT) Protocol data transmission protocol and of the SMTP data transmission protocol is almost equal, but when the amount of data is large, the UDP-based Data Transfer (UDT) Protocol data transmission protocol takes more time than the SMTP protocol. The experimental results show that the implementation of the UDP-based Data Transfer (UDT) Protocol data

Table 1. Data fusion results

Sensor number	Date	Average temperature/degrees	Pressure /kpa	Thermal power generation /MW	Photovoltaic power generation /MW
1	July	24.7	42	57	/
2	July	25.7	50	42	/
3	July	23.9	46	62	/
4	July	25.6	38	73	/
5	July	23.3	40	51	/
6	July	25.9	41	162	/
7	July	29.6	48	/	50
8	July	30.1	39	/	42
9	July	29.5	44	/	33
10	July	28.4	43	/	25

transmission protocol features a high efficiency of energy industry Internet mass data transmission, which can significantly reduce the time from data collection to its storage and contribute to the rapid storage of energy industry Internet data.

In order to verify the data storage results for this method, the internet information for the energy industry in the chosen city for the timespan between 2020 and 2024 was obtained by using the access interface provided by the electric utility company, and the cluster storage was carried out by using this method. The storage results for a part of this information are shown in Figure 8.

MDS	Photovoltaic power generation	Thermal power generation	Hydroelectric power generation	Other data	Admin
Equipment distribution	Photovoltaic lines				
Photovoltaic site	Partition list	Daily power generation	Total power generation	Power	Other
Photovoltaic circuit	Route	Unit	Power (kW)	Generating capacity (kWh)	Total power generation (WkWh)
Inspection record	Route 3	Unit 1	12532	12532	60153.6
Other	Route 3	Unit 2	12532	12532	60153.6
	Route 3	Unit 3	12532	12532	60153.6
	Route 3	Unit 4	12532	12532	60153.6
	Route 4	Unit 1	12603	12603	60201.8
	Route 4	Unit 2	12603	12603	60201.8
	Route 4	Unit 3	12603	12603	60201.8
	Route 4	Unit 4	12603	12603	60201.8
	Route 4	Unit 5	12603	12603	60201.8

Figure 8. Partial data storage results

Figure 8 shows certain data from the photovoltaic power plants in the selected city, including the area, the generated electricity and other precise information. By using this method, the power generation situation of the power plants in any area of the selected city can be clearly known, and the maintenance time of the power plants can be determined, so that instructions can be issued in a targeted manner to improve the service life of the power plants.

To evaluate the advantages of the method presented in this paper in terms of storage speed, the following five existing methods were selected for a comparative experiment:

- (Dharminder et al., 2022): A signature encryption storage scheme based on post-quantum integer lattices;
- (Vishwakarma et al., 2022): A cloud storage method based on resource scheduling optimization in a secure system;
- (Maftei et al., 2023): An IoT massive data storage scheme based on blockchain technology;
- (Ye & Park, 2021): A trusted data storage method combining blockchain and IPFS;

- (McManamon et al., 2021): A high-capacity and high-speed data storage method based on laser communication.

The experiment was conducted based on the data storage time as the indicator, and the results are shown in Figure 9.

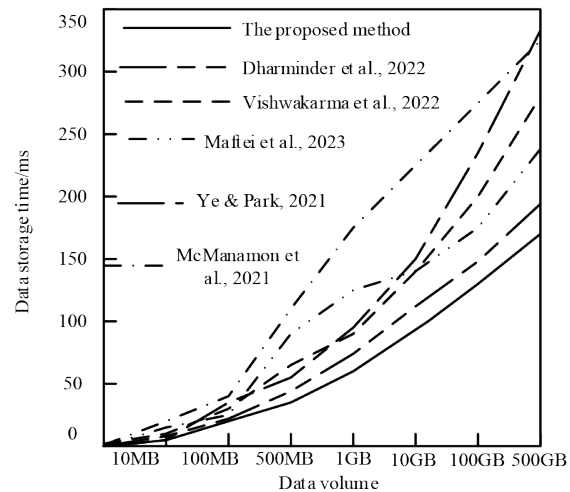


Figure 9. Data storage time

According to the data storage time comparison results shown in Figure 9, the storage time for all methods increases as the data volume increases, but the growth rate and stability vary significantly. The storage time fluctuation for the method of Maftei et al. (2023) is the highest. When the data volume exceeds 200GB, the storage time increases sharply, mainly due to the decreased data writing efficiency caused by the blockchain consensus mechanism and block generation delay. The method of Dharminder et al. (2022) features a relatively high overall storage time because it performs complex Galois encryption and signature verification on each data packet before storage, resulting in a high computational overhead. The method of Ye & Park (2021) uses IPFS for a distributed storage, but content addressing and database sharding storage require additional hash calculations and network lookups, leading to a higher latency. The method of McManamon et al. (2021) is limited by the existing hardware model in the simulation environment, with optical signal conversion and error correction processing adding additional delays. Finally, the method of Vishwakarma et al. (2022) shows a relatively stable performance, with the storage time being second only to the one obtained by the proposed method, but it still needs to transfer all raw data to the cloud, and network latency and centralized storage bottlenecks limit its efficiency.

To sum up, the proposed method obtained the shortest data storage time for all data volumes, especially when handling data larger than 500GB. This result is attributed to the end-edge-cloud collaborative architecture: the edge computing layer performs fusion preprocessing on multi-source data, significantly reducing the amount of data to be transmitted; the UDP-based Data Transfer (UDT) Protocol as a transmission protocol has a dynamic congestion control mechanism to ensure the efficient and reliable transmission of large data blocks; the CFSlight cluster file system features a storage structure based on separating metadata from data, achieving high concurrent reading and writing; the dynamic load balancing strategy monitors the server response time in real time to avoid single-point overload. The multi-level optimization enables this method to maintain the highest efficiency when storing large-scale energy industry Internet data.

The algorithm presented in this paper is also employed to balance the servers, and the load distribution among the servers in the cluster is shown in Table 2. As can be seen from Table 2, after the load balancing algorithm is implemented, the load levels of all servers remain below their respective load warning lines. Although servers No. 1 and No. 7 have special requirements, their loads are also kept within the warning limits. To sum up, it can be seen that the load balancing method employed by this algorithm is effective and can prevent a single server from being full.

4. Conclusion

This paper addresses the massive data storage requirements of the energy industry Internet of Things and proposes an information cluster storage algorithm that would enable the access of edge devices. This algorithm generated a four-layer storage framework, it performed fusion

preprocessing on multi-source sensor data at the edge computing layer, it used the UDP-based Data Transfer (UDT) Protocol to achieve an efficient and reliable data transmission, and completed data storage based on the CFSlight cluster file system. At the same time, a dynamic load balancing strategy is introduced to optimize the performance of the server cluster. The contributions of this paper are as follows:

1. In the data fusion stage, the algorithm can effectively integrate the operation data of 10 different sensors and output a clear and structured fusion result, laying the foundation for subsequent data storage;
2. In terms of data transmission, the UDP-based Data Transfer (UDT) Protocol protocol has reached a significantly higher data transmission efficiency than the HTTP and TCP/IP protocols when handling data volumes of over 500GB, effectively reducing the overall time from data collection to storage;
3. As regards the storage performance comparison, the storage time for the proposed method is always lower than that of the other employed methods, especially when dealing with data volumes of over 500GB, where its advantage is more prominent. This is due to the data preprocessing at the edge computing layer which reduces the amount of data to be transmitted, to the dynamic congestion control of the UDP-based Data Transfer (UDT) Protocol ensuring transmission efficiency, and to the metadata and data separation architecture of the CFSlight cluster file system achieving a high concurrent reading and writing;
4. In terms of load balancing, the dynamic load balancing strategy keeps the load of 7 servers in the cluster below the warning line, effectively avoiding a single-point overload and improving the stability and scalability of the system.

Table 2. Server cluster loading results

Server number	Loading degree/%	Load warning/%	Enabled state	The highest loading ever recorded	Remark
1	57	60	On	67	The primary server loading is less than 60%
2	69	75	On	92	/
3	72	75	On	85	/
4	70	75	On	80	/
5	73	75	On	97	/
6	71	75	On	80	/
7	80	85	On	99	Store other data

REFERENCES

- Achour, S., Benattou, M. & Lanet, J. L. (2021) Formal model of conformity analysis method of java bytecode programs annotated with bml specifications. *IAENG International Journal of Computer Science*. 48(2), 343-355.
- Alves, F., Mateus-Coelho, N. & Cruz-Cunha, M. (2021) Encryption file system framework - proof of concept. *Procedia Computer Science*. 181(1), 1237-1246. <https://doi.org/10.1016/j.procs.2021.01.322>.
- Bi, J., Niu, S., Yuan, H. et al. (2025) Energy-minimized partial computation offloading in satellite-terrestrial edge computing networks. *IEEE Internet of Things Journal*. 12(5), 5931-5944. <https://doi.org/10.1016/j.ijleo.2025.172321>.
- Chen, B., Liu, X., Xu, H. et al. (2025) BSSN: enabling adjustable blockchain storage for resource-constrained IoT scenarios. *IEEE Internet of Things Journal*. 12(4), 4262-4274. <https://doi.org/10.1109/JIOT.2024.3483898>.
- Deji, R., Choudhary, B. C. & Sharma, R. K. (2022) Hydrogen sulfide gas sensor using osmium doped graphene nanoribbon: an insights from DFT study. *Materials Letters*, 306(1), Art. No. 130986. <https://doi.org/10.1016/j.matlet.2021.130986>.
- Dharminder, D., Kumar, U., Das, A. K. et al. (2022) Secure cloud-based data storage scheme using postquantum integer lattices-based signcryption for IoT applications. *Transactions on Emerging Telecommunications Technologies*. 33(9), 4540-4567. <https://doi.org/10.1002/ett.4540>.
- Fabian, B., Bender, B., Hesseldieck, B. et al. (2020) Enterprise-grade protection against e-mail tracking. *Information Systems*. 97(3), Art. No. 101702. <https://doi.org/10.1016/j.is.2020.101702>.
- Kaliyamoorthy, P. & Canessane, R. A. (2021) Enhancing data security using global mutation based novel artificial immune network optimization in public cloud storage system. *Transactions on Emerging Telecommunications Technologies*. 33(1), Art. No. e4390. <https://doi.org/10.1002/ett.4390>.
- Maftai, A. A., Lavric, A., Petrariu, A. I. et al. (2023) Massive data storage solution for iot devices using blockchain technologies. *Sensors*. 23(3), Art. No. 1570. <https://doi.org/10.3390/s23031570>.
- McManamon, P., Vedadi, A., Willner, A. E. et al. (2021) High capacity and access rate, data storage using laser communications. *Optical Engineering*. 60(1), Art. No. 015105. <https://doi.org/10.1117/1.OE.60.1.015105>.
- Orowski, C., Cofta, P. & Orlowski, A. (2022) The rule-based model of negentropy for increasing the energy efficiency of the city's digital transformation processes into a smart city. *Energies*. 15(4), Art. No. 1436. <https://doi.org/10.3390/en15041436>.
- Ovesen, A. B., Nordmo, T.-A. S., Johansen, H. D. et al. (2021) File system support for privacy-preserving analysis and forensics in low-bandwidth edge environments. *Information*. 12(10), Art. No. 430. <https://doi.org/10.3390/info12100430>.
- Prakash, V., Savaglio, C., Garg, L. et al. (2022) Cloud- and edge-based ERP systems for industrial Internet of Things and Smart Factory. *Procedia Computer Science*. 200, 537-545. <https://doi.org/10.1016/j.procs.2022.01.251>.
- Sain, C., Banerjee, A., Biswas, P. K. et al. (2021) Sensor angle-based control strategy and dynamic analysis of a sinusoidal pulse width modulation-operated permanent magnet synchronous machine drive for electric propulsion unit. *International Transactions on Electrical Energy Systems*. 31(12), Art. No. e13090. <https://doi.org/10.1002/2050-7038.13090>.
- Sumathi, M. S. & Anitha, G. S. (2020) Link aware routing protocol for landslide monitoring using efficient data gathering and handling system. *Wireless Personal Communications*. 112(4), 2663-2684. <https://doi.org/10.1007/s11277-020-07169-6>.
- Sun, J. & Wan, Z. (2022) Application of configuration algorithm of carbon nano tube yarn sensor in health monitoring. *Ordinance Material Science and Engineering*. 45(2), 129-133.
- Tsumachi, N., Shibayama, M., Kobayashi, R. et al. (2021). Demonstration experiment of a 5G touchless gate utilizing directional beam and mobile edge computing. *IEICE Transactions on Communications*. E104/B(9), 1017-1025. <https://doi.org/10.1587/transcom.2020FGP0001>.
- Vishwakarma, S., Goswami, R. S., Dutta, S. et al. (2022) RETRACTED: Cloud data storage with improved resource scheduling in healthcare application based on security system. *Optik*. 272(2), Art. No. 170225. <https://doi.org/10.1016/j.ijleo.2022.170225>.
- Ye, H. & Park, S. (2021) Reliable vehicle data storage using blockchain and IPFS. *Electronics*. 10(5), Art. No. 1130. <https://doi.org/10.3390/electronics10101130>.
- Zhang, R., Hallström, S., Mao, H. et al. (2021) Accurate viscosity prediction for molten slags: a new model and database. *ISIJ International*. 61(5), 1379-1388. <https://doi.org/10.2355/isijinternational.ISIJINT-2020-586>.



This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.