

DESIGN AND IMPLEMENTATION OF BIG DATA PLATFORM IOT DATA ACCESS ARCHITECTURE

*Hongfei Xiao
Wenwen Li
Shiqi Tang*

To solve the problems related to the equipment of the joint platform for industrial product data collection, an Internet connectivity structure has been developed, which is divided into three levels: equipment and facilities, network access center and application services. Within this framework, we have created a standard engine, hardware centre, protocol centre, information model centre, isochronous sequence, facilities, and data access. Through a big data platform project, we have also realised the network architecture and Internet connection function module. The application demonstrates that the system's structure offers a reliable method for accessing isomers on the big data platform, offers a foundation for the study of data consolidation platforms, and permits secure access to a significant amount of data within a single functional interface. According to the testing findings, this technology outperforms conventional methods by 12% in terms of access performance, transmission delay, and data throughput.

Keywords: big data platform, IoT connection, Data access architecture, NB-IoT

INTRODUCTION

The primary objective of each group's daily operation and maintenance is to lower the rate of equipment failure due to the dispersed and numerous pieces of equipment. However, in the equipment process, it is impossible to accurately quantify the loss level of asset and equipment efficiency, effectively identify the key points to improve asset and equipment efficiency, and it is also challenging to correctly evaluate and effectively motivate the production team without information-based data support and a scientific measurement index system.

Data interconnection is achieved through the reconstruction of equipment operation, monitoring, management, maintenance, overhaul, and other links. This integration of big data and cloud computing into various technical links of data and traditional industries is expected to realise the intelligent operation, maintenance, and management of on-site stations [Wang et al. 2020, Pääkkönen & Pakkala, 2015].

Smart production is based on digitization, informatization, and standardisation, with management and control integration, the Internet of Things (IoT) as the platform, digital twin technology as the support, elastic resource configuration, and heterogeneous computing as the core task. Smart production effectively integrates computing, storage, and networks, and forms a full level open architecture fusing edge and cloud through cross-border integration. Improve intelligence consistently in order to provide a more amiable, secure, efficient, and dependable energy source [Cai et al. 2016, Verma et al. 2017].

The intelligent platform architecture, which offers standardised access to hardware level data on massive data systems, includes intranet connections as a key component [Ray, 2016, Khan et al. 2015]. This research develops a functional modular network connection scheme, examines the technical benefits of network connection, and offers a network connection scheme for a big data platform [Qui et al. 2018, Mohammadi et al. 2018]. The network connection module was developed on the large cloud data centre platform to realise the large data platform for the group data set access centre, effectively validating the program's viability [Mahdavinejad et al. 2018, Iqbal, Zerguine & Khan 2021].

The paper is organized as follows: §2: Introduction to related work and theory. §3: The main research methods and innovation model of this paper. §4: Analysis of the proposed method as well as the model. §5: Summary of the article.

RELATED WORK

The term IoT was first proposed by Professor Ashton of MIT in 1999, which refers to the connection between things and people [Iqbal et al. 2020]. Using the Internet and other communication technologies, connect sensors installed on different objects in a new way to establish a new network, that is, a new network, which can be installed on different objects [Tanwar et al. 2022]. The IoT refers to the use of a series of sensor devices to collect data in real time through RFID technology and wireless data communication technology, to achieve intelligent control of objects [Badshah et al. 2022]. Its essence is to exchange information between objects and people. Its main features can be summarized as universal perception, reliable communication and intelligent processing [Iwendi & Wang, 2022].

As in Figure 1, the IoT network mainly includes sensors, two-dimensional code, radio frequency technology, etc. The function of the network layer is to transmit the collected data and instructions through the LAN and WAN[Zhou et al. 2023]. The application layer is the interface between the network and users. It provides intranet based applications and intelligent management in all areas[Lou, Zhang & Bai 2023].

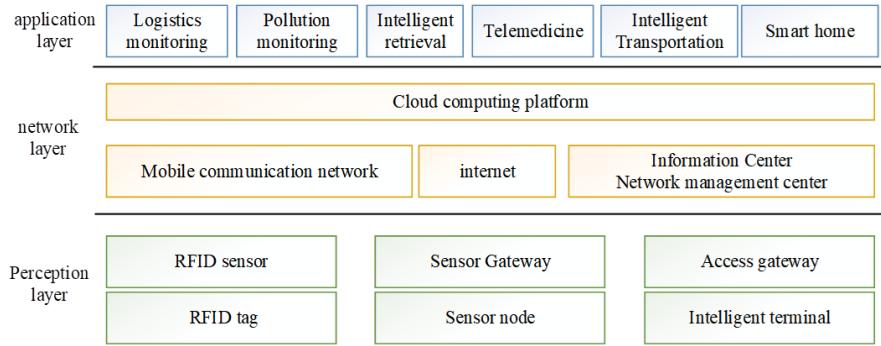


Figure 1: Structure of IoT

With the continuous maturity of the IoT technology and the emergence of various intelligent devices, we will enter an era of interconnection of all things, and the IoT market will become larger and larger [Ali et al. 2023]. However, the IoT we really use is very few, such as smart home, Internet of Vehicles, etc. Although we have this concept for a long time, it does not benefit the public. The industry summarizes that there are three main reasons that limit the development speed of the IoT: first, edge computing is a difficult problem. Due to sensor, local decision-making, multi-protocol compatibility and other reasons, most of the IoT projects have not progressed to the stage of proof of concept. Second, there is a lack of mature IoT platform. What we often see are just colorful widgets, icons, and other visual creative interfaces that suppliers show us. There are few IoT platforms that really provide effective solutions. Third, the combination of software and hardware. The manufacturers of IoT sensors and gateways lack strong alliances with software suppliers.

Three levels make up the Internet of Things: the application layer, the network layer, and the perception layer. A good basis for the development of the IoT can be built with a scalable platform, a reliable network, rich applications, synchronous development, and organic composition. The IoT platform and its associated devices serve as the IoT system's connectivity channel. The IoT lacks application flexibility without it.

At this stage, the domestic IoT platform manufacturers are mainly divided into four fields: communication field (operators, communication equipment manufacturers), Internet field (BAT), software system service field (IBM, Microsoft) and vertical field (smart cloud). Each IoT manufacturer has its own production standard, which is not resource sharing, leading to serious fragmentation. This fragmentation will worsen as additional applications are added. Therefore, in order to address a variety of issues, a uniform standard and a unified platform are required. The IoT platforms being investigated right now are all application platforms, which are exclusively useful in a single vertical industry. All application platforms can use the IoT access platform that was the subject of this article, which accelerates and shortens the R&D process.

METHODS

Construction structure

This section describes the framework for implementing IoT by looking at real-world research projects, IoT system theory and technology, and pertinent IoT examples. The elements of the IoT's structure are shown in Figure 2 and include: coverage for industry applications, network connections, access levels, network layers, and sensors.

In this study, the device management platform of the application layer is referred to as the IoT connection management platform, and Figure 3 shows its general architecture. Large IoT platforms, such as the IoT access

ecosystem headed by AWS, Google, and Microsoft, as well as the IoT management platforms of large telecom operators, like the IoT platform from China Telecom and the "One Net IoT platform" from China Mobile, are already available. The Tiangongkai IoT platform from Baidu, the AWS IoT service platform from Amazon, and the Ocean Connect platform from Huawei Technologies are examples of large technological companies.

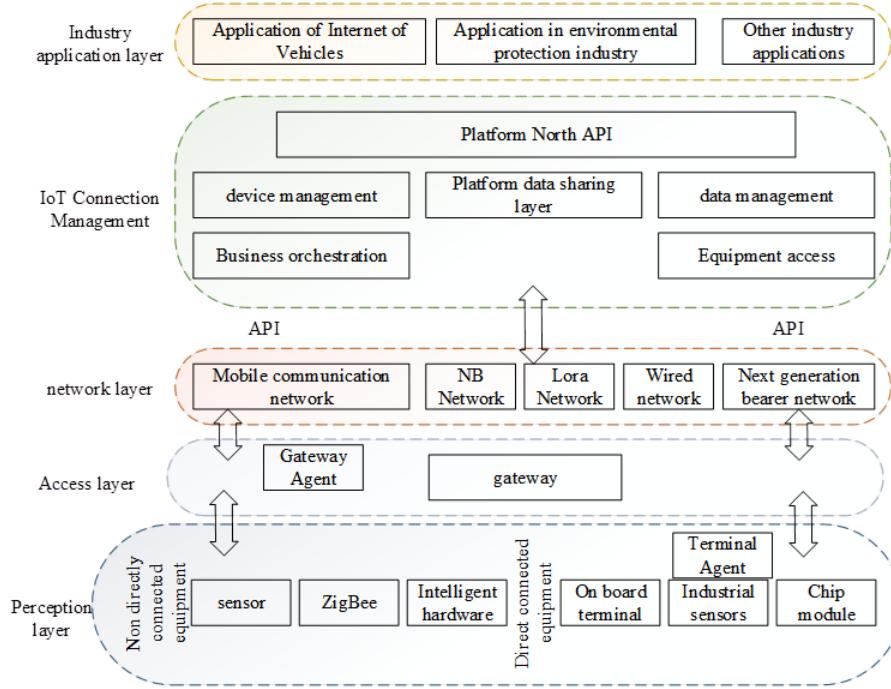


Figure 2: Architecture Design of IoT Based on Device Access

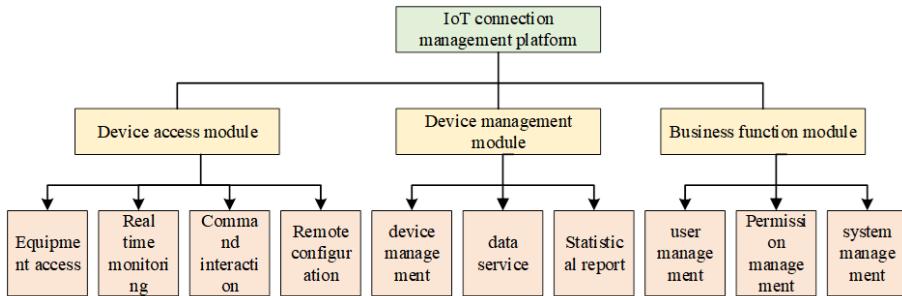


Figure 3: Overall architecture of application layer IoT connection management platform construction

IoT equipment access management platform

As shown in Figure 4, the architecture of the IoT equipment access management platform is primarily designed in this area. We can see that the IoT platform must address two issues in order to provide unified access to and administration of terminal devices: Build an equipment management platform from the platform layer to address the issues of equipment access, equipment management, and visualisation. Resolve the unified access problem of terminal devices from the IoT perception layer.

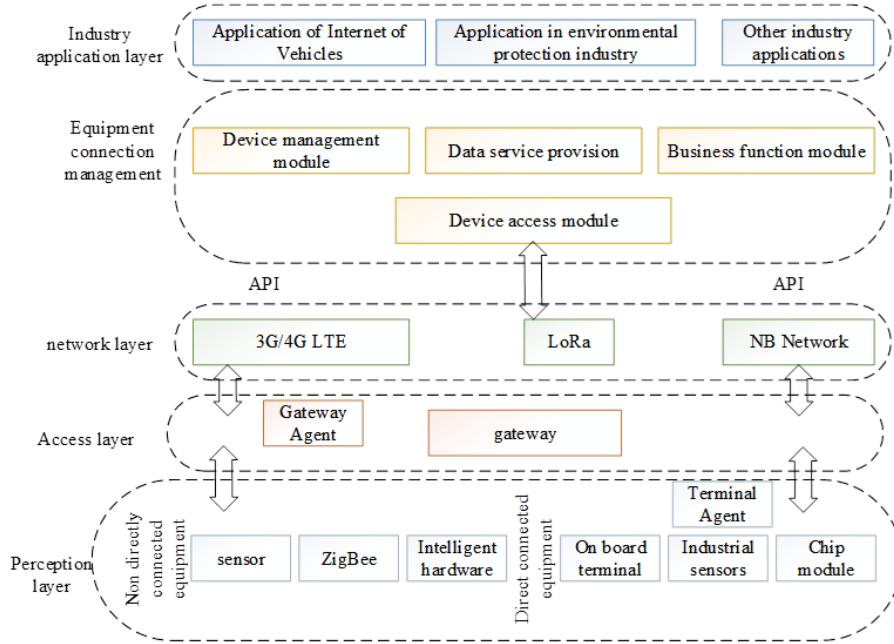


Figure 4: Overall architecture of IoT equipment access management platform

(1) Perception layer

Direct linked devices, commonly referred to as IP devices, are objects with specific capabilities, the ability to generate data, and the ability to interact with the outside world. Vehicle terminals, industrial sensors, residential terminals, and other terminal devices having network connectivity capabilities are examples of typical gadgets. To implement data reporting and receiving processes, the gateway is required. The term "indirect connection equipment" refers to a class of devices that mostly consists of sensors, ZigBee, and some intelligent hardware.

(2) Access layer

The IoT access layer endows the sensing layer devices with network access capabilities and unifies data and protocols through the Agent gateway. The resource information can be collected and reported through the gateway agent or terminal agent. It can realize the complete networking and remote control of access devices.

The 4G low-power IoT and other communication modules can be used to access the network for equipment without direct IP connection capability, and the gateway software can be utilised to accomplish intelligent access. The gateway agent is used with devices that have IP capabilities. To customise the abstract gateway agent for the device, the terminal device incorporates the gateway agent SDK. The terminal device must be compatible with the agent SDK's operating environment.

(3) Network layer

The network layer is the medium for remote information interaction between terminal equipment and IoT connection management platform, and the corresponding transmission mode can be selected according to different application scenarios. Wireless network access mode is mainly adopted here, including 3G/4G LTE and low-power NB IoT network communication.

(4) Industry application layer

After collection and processing, it provides the industry application layer with data analysis or application

system development, effectively uses the data, and combines big data and AI related technologies to achieve intelligent, accurate and scientific industry application platforms.

NB IoT equipment access scenario

The access to terminal devices in the NB IoT low-power scenario is designed and implemented in this section. It can be used on the platform to generate commands and report small hardware packets due to its low power and slow speed. It has been extremely helpful in lowering the price of communication equipment, incorporating new technology into the equipment access network for research and development, and assuring its effectiveness. It can be used to send monitoring data or huge data documents via 4G. Figure 5 depicts their relationship with one another.

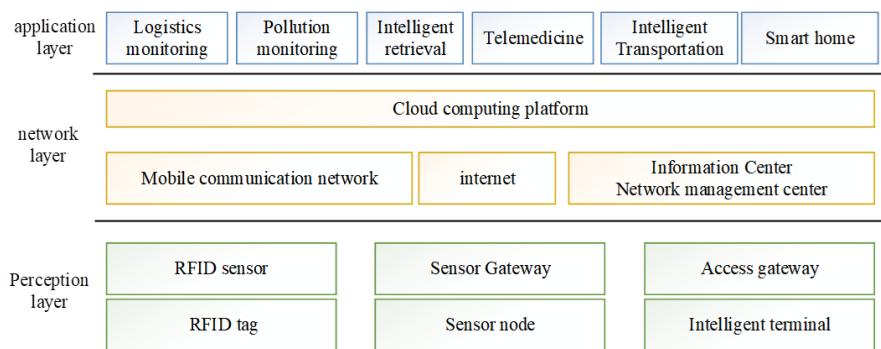


Figure 5: NB IoT Vertical Industry System Architecture

In this paper, intelligent gateway based on Zyngq7000, mobile BC95NB communication module, and external IoT sensor equipment are used to complete the access of terminal IoT equipment in the low-power IoT scenario. The remote BC communication module includes the USIM card and RF antenna of the telecom operator, BC95-TE-A and the two USB ports on the left. Below is the device startup switch.

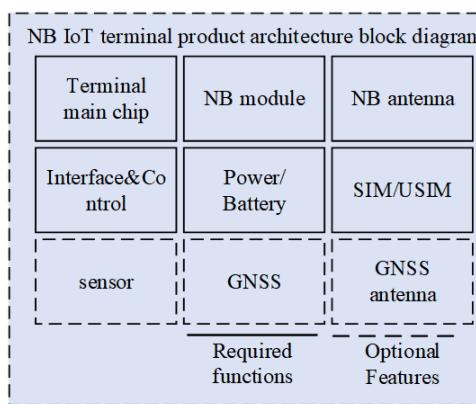


Figure 6: Product Architecture of NB IoT Module

Figure 6 shows the NB IoT module products' architecture. Depending on the intended use, the dotted line function can be individually set. Green function modules, NB modules, SIM/USIM, and NB antennas must be added to existing hardware in order to establish NB IoT networking. The power supply (including voltage drop and LDO) must meet pertinent requirements for ripple noise, dynamics, loop stability, etc.; the clock must satisfy the necessary requirements for frequency offset and phase noise; and the RF must satisfy the

necessary requirements for system transmission power, EVM, receiving sensitivity, isolation, noise figure, and other pertinent indicators.

The gateway will be connected to the NB communication module once the equipment and gateway pass the combined commissioning test. The remote BC95 communication module is used in this section. In order to complete device registration and activation, complete docking with the platform, and submit sensor device data gathering through the gateway, the communication method requires AT instructions. The platform can manage and operate the equipment as well as display the information related to the equipment and data. The IoT platform is connected to the NB module, which serves as the communication module. Finally, the data collected by the equipment can be obtained through the platform's open API and integrated into the Alibaba Cloud IoT equipment management platform to achieve the equipment access in the NB IoT scenario, see Figure 7 for the detailed access design process.

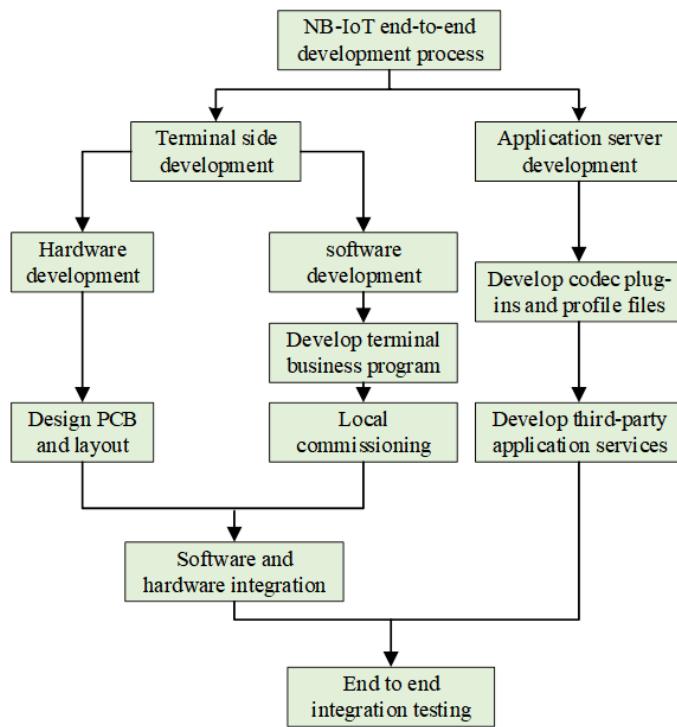


Figure 7: NB IoT Development Access Process

At present, the terminal controls the communication module through AT command, so the terminal manufacturer not only develops its own business functions, but also needs to develop a scheduler to call the AT command control device and external communication module, as in Figure 8.

The NB IoT module connects to the terminal's network, sets up SSCOM on the computer, unlocks the serial port, and recognises the network access procedures. You must set the address of the docking IoT platform and reset the terminal before each network access. Signal detection is done after the function switch is turned on. A signal that is stronger than 16 indicates that the device can communicate. Enter AT command: AT+CGATT=1 to activate the network access. If OK is returned, it indicates that the activation is successful. The module can start dual communication with the NB core network platform. Before communication, the uplink and downlink data notification function need to be set to prevent missing data transmission or reception.

The device can send data to the platform through the NB module and can also receive messages from the platform. Therefore, the next interview is to receive and display these messages on the platform, obtain these

data in real time through the open API on Ocean Connect platform, and store them in the built Alibaba Cloud device management platform.

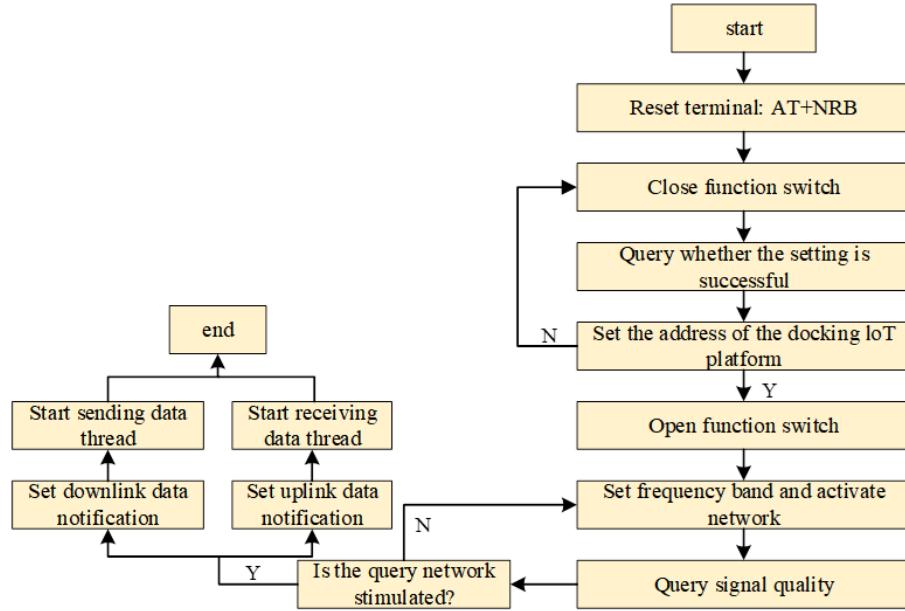


Figure 8: Flow Chart of NB Module Communication Module

Heterogeneous data access

The data accessed to the big data platform is mainly divided into equipment real-time data (time series data, such as data, photovoltaic production real-time data), object data (such as fault recorder, vibration file, image and video) and relational data (such as production management data). According to different data types, different front-end services need to be used in the data collection process, such as timing data front-end, fault log front-end, structured data front-end, API interface front-end, etc.

The timing data linked to the big data platform includes high temporal continuity, enormous throughput, and fluctuations like peak value and latency. The single machine must therefore have a high throughput and ensure that no records are lost or duplicated. Continuous access to timing data will not be impacted by single point failure. The data service platform offers three access options for timing data, as shown in Figure 9, for various usage scenarios: real-time access, batch access, and timed batch access. The real-time access mode primarily focuses on the monitoring data produced in real-time by the device sensors, which must be continually retained for processing in the future.

The object data accessed to the big data platform mainly includes video monitoring data and log text data. Object data registration, MD5 verification, and object upload are all parts of the access process for object data. Data access standards are the foundation for object data registration, which includes the addition of descriptive data for object data, such as file name, type, and purpose, to create object meta information and streamline retrieval applications. To guarantee the integrity of object data, MD5 verification is dependent on the quality of the data. The object upload is based on a data storage approach that uses object data storage and the distributed file system write interface. The IOT connection extract transform load (ETL) tool offers three access services for the application scenarios of various object data access: upload using the management console interface, upload using Rest API programmes, and upload using Java SDK programmes. The upload operation using the management console interface is simple, but it cannot process many files; Uploading with

Rest API program is friendly to web application development and is suitable for uploading medium size files; Use the Java SDK program to upload scenarios suitable for large-scale file upload.

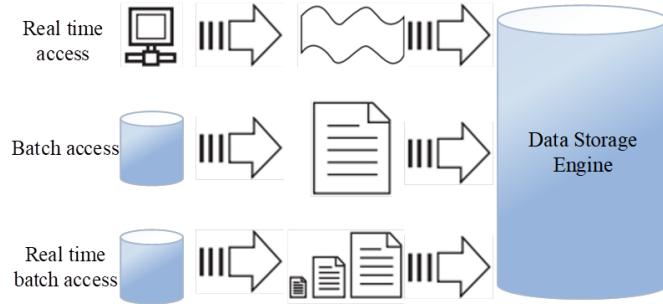


Figure 9: Time sequence data access

Communication mechanism

Figure 10 shows the communication system between the IoT platform and equipment. The IoT platform enables users to monitor logs, adjust configurations, add instructions, access devices, and do other tasks. Users on the IoT platform carry out the device management function via the management interface, carry out pertinent operations on the interface, start the platform middleware's Socket server function, and transmit pertinent commands to the server listener on the device management platform. The server delivers the pertinent commands to the device access module, which is in charge of examining and handling the user's requests, after receiving them.

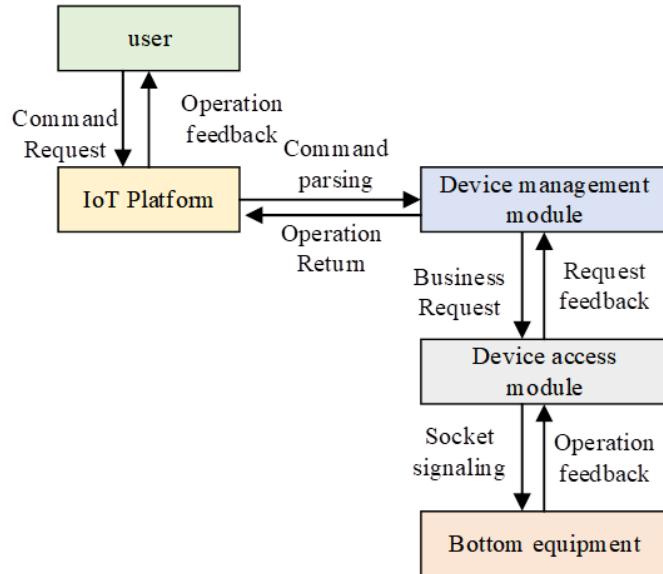


Figure 10: Communication Mechanism between IoT Equipment and Platform

EXPERIMENTS

The client transmits a lot of data to the server in a client-server system architecture. The performance of the access platform's concurrent access and response time is crucial to the platform's availability and stability. RT stands for the system's response time, or how quickly it responds to queries. The response time is typically good and falls within 100ms. The amount of simultaneous connections made by regular users that the system can support is referred to as a concurrency scenario. Concurrency scenario testing helps evaluate the system's carrying capacity and delay performance.

Start a client, deploy the JDK1.8 environment, configure the jar package of Netty4.2, send Tcp client connections through Java's ScheduledExecutorService, and accumulate the final time of connecting all these requests on the server. The initial value of concurrent requests is 1000. Each time the client sends a connection, it prints it. After the middleware server is started, it receives the Tcp request from the remote client, and prints the IP and port number of the corresponding client each time it executes, as in Table 1.

Table 1: Connection Information of IoT Access Platform Server

boot. server. TcpServer handler	Connected client address: XXX
boot. server. TcpServer handler	Connected client address: XXX
boot. server. TcpServer handler	Connected client address: XXX
boot. server. TcpServer handler	Connected client address: XXX

System evaluation

The performance of the server is verified through the concurrency test under the middleware server simulation environment. The concurrent response time of the server is in Table 2, where C1, C2, and C3 are the set number of cores in different thread pools. The high concurrency needs of the current case can be satisfied by middleware's high concurrent connection processing performance, which has a response time under 100000 connections of less than 100ms. The latency in the simulation test environment becomes too high whenever there are one million connections active at once. Distributed deployment is a solution we can use to address this issue. The distributed method can manage more orders of magnitude more concurrent access in the face of massive concurrency. The platform may work together and process distributed transactions using Nginx's request dispersion. Table 3 displays the number of transactions executed each second across various threads. The test results show that when the concurrency is 1, 360 transactions are processed per second. In the case of single thread, the ability to process transactions is limited. When the number of threads is set to 100, the TPS is about 3400, which means the number of transactions processed per second is 3400. It can be seen that with the increase of concurrency, the transaction processing ability of the middleware server is enhanced. It also has significant improvement and can meet the requirements of concurrent access of lightweight platform equipment.

Table 2: Concurrency Scenario Response Time

[UNK][UNK];	100	1000	2000	5000	10000	100000	1000000
C1(ms/1)	9	8	11	17	16	49	325
C2(ms/10)	10	13	13	18	27	58	304
C3(ms/20)	14	17	22	29	33	68	312

Table 3: TPS Test Results under Load Data

Serial No	Message (bytes)	Number of threads	TPS
1	2048	1	361
2	2048	9	1601
3	2048	33	2101
4	2048	100	3413
5	2048	500	4202

There are 500 million pieces of test data collected by terminals of three types of equipment. The access performance results are in Figure 11 and Figure 12. The horizontal axis in the figure indicates the number of threads, the vertical axis in Figure 13 indicates the throughput (times/second), and the vertical axis in Figure 14 indicates the delay time (milliseconds).

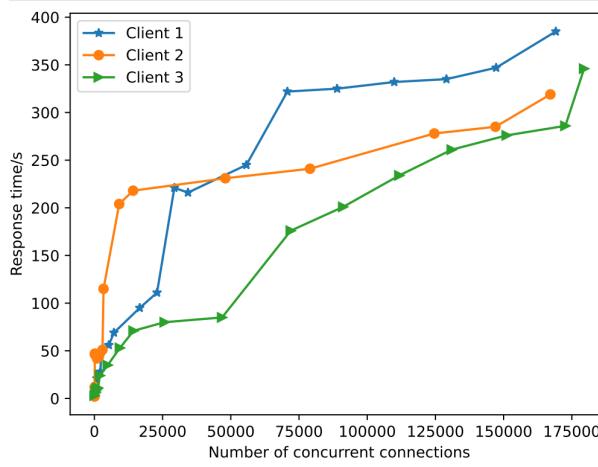


Figure 11: Concurrent Access Response Test Results

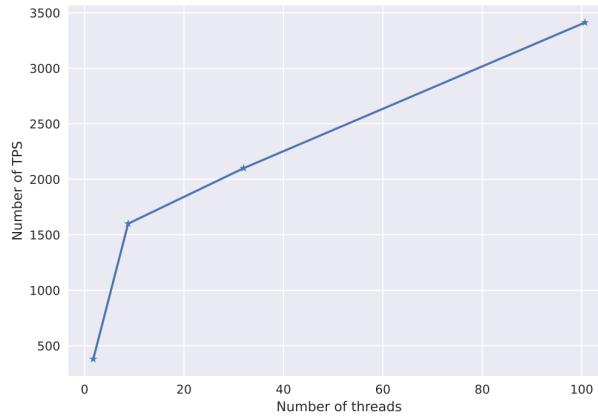


Figure 12: TPS Testing

In Figure 13, with the increase of the number of threads, the system throughput will continue to increase. When it rises to 1000, the system throughput will change very little. With the increase of the number of threads, the system bandwidth resources are basically exhausted, so the throughput will not increase. Therefore, our system and traditional system have basically the same changes to the throughput.

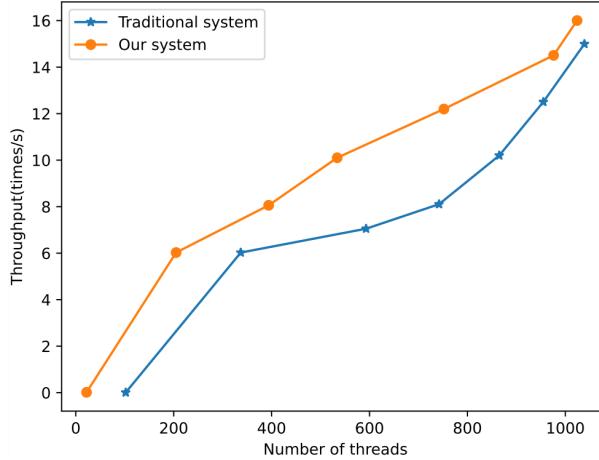


Figure 13: Performance Throughput Test Curve

In Figure 14, with the increase of the number of threads, CPU resource scheduling is frequent, and the CPU resource allocated by a single thread will continue to decrease, so the access delay will continue to increase. However, the data model trained by random number generator and C4.5 algorithm in this paper can accurately locate the device partition and avoid hot spot problems, which can reduce the resource scheduling frequency of CPU, so the access performance delay will be improved compared with the traditional system.

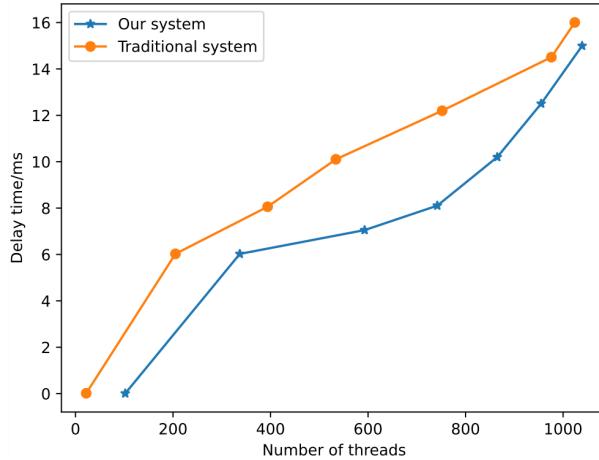


Figure 14: Performance Delay Test Curve

TCP data receiving module is an important module of the platform, which is responsible for receiving and preprocessing sensor data, and directly determines the network communication performance of the platform. This section will explore the specific performance and optimal configuration of TCP servers through experimental design. In the experiment, when n is small, it increases by 10 times each time, and when n is large, it slows down the increase. Select 1, 10, 100, 200, 500, 600, 700, 800, 900, 1000. To minimize the time of establishing data connection and the impact of server fluctuation, each test takes 30s. The test results are in Table 4.

With a rise in concurrency, the server's data processing rate gradually drops during the transmission period (100 ms) 42. Less than half of the data will be handled in time as the concurrency grows by 1000. Therefore, it is preferable to keep server concurrency within 500 in the current unfavourable situation. The sensor data acquisition frequency, however, seldom approaches the level of 100 ms and thousands of concurrent values at

the same time in the actual generation environment. The theoretical concurrent quantity reaches 5000 when the acquisition frequency * increases to the level of 1s. In addition, the "data processing ratio *" parameter selected in the wood test is not particularly appropriate. It describes the processing ratio of "same time". In the actual cattle production environment, the 100ms delay is acceptable. At this time, in theory, the concurrency can also be doubled to 5000 stone. To sum up, although the performance of TCP servers will gradually decline with the increase of concurrency, it is enough to cope with most actual production environments.

Table 4: Server Test Results

Test No	Concurrency n	Send data (Byte)	Server returned data (Byte)	Data processing ratio P
1	1	799	799	100.00%
2	10	8446	8446	100.00%
3	100	81298	81279	99.98%
4	200	162058	159925	98.69%
5	500	264517	246391	93.15%
6	600	251554	226369	89.99%
7	700	275305	202109	73.42%
8	800	277339	196624	70.89%
9	900	424435	219448	51.71%
10	1000	2118259	1002175	47.32%

CONCLUSION

(1) The IoT connection architecture on the big data platform can work with data security protection to offer a secure and dependable data transmission route. (2) The IoT connection architecture can facilitate data collection and access of big data platform and shield the differences of underlying equipment. (3) The equipment, communication status, and operation of the side platform and central platform of the data cooperation architecture are made easy and unified by the IoT connection architecture.

ACKNOWLEDGMENTS

This study was financially supported by the Information security and big data application technology innovation platform(Project No.: YJP-2019-01);Internet plus Urban and Rural Creative Development Center(Project No.: YJP-2021-03).

CONSENT FOR PUBLICATION

All authors reviewed the results, approved the final version of the manuscript and agreed to publish it.

DATA AVAILABILITY

The experimental data used to support the findings of this study are available from the corresponding author upon request.

CONFLICTS OF INTEREST

The authors declared that they have no conflicts of interest regarding this work.

REFERENCES

Wang J, Yang Y, Wang T, Sherratt RS, Zhang J. Big data service architecture: a survey. *Journal of Internet Technology*. 2020 Mar 1;21(2):393-405.

Pääkkönen P, Pakkala D. Reference architecture and classification of technologies, products and services for big data systems. *Big Data Research*. 2015 Dec 1;2(4):166-86.

Cai H, Xu B, Jiang L, Vasilakos AV. IoT-based big data storage systems in cloud computing: perspectives and challenges. *IEEE Internet of Things Journal*. 2016 Oct 19;4(1):75-87.

Verma S, Kawamoto Y, Fadlullah ZM, Nishiyama H, Kato N. A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials*. 2017 Apr 14;19(3):1457-77.

Ray PP. A survey of IoT cloud platforms. *Future Computing and Informatics Journal*. 2016 Dec 1;1(1-2):35-46.

Khan Z, Anjum A, Soomro K, Tahir MA. Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing*. 2015 Feb 18;4(1):2.

Qiu T, Chen N, Li K, Atiquzzaman M, Zhao W. How can heterogeneous internet of things build our future: A survey. *IEEE Communications Surveys & Tutorials*. 2018 Feb 8;20(3):2011-27.

Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*. 2018 Jun 6;20(4):2923-60.

Mahdavinejad MS, Rezvan M, Barekatain M, Adibi P, Barnaghi P, Sheth AP. Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*. 2018 Aug 1;4(3):161-75.

Iqbal N, Zerguine A, Khan S. OFDMA-TDMA-based seismic data transmission over TV white space. *IEEE Communications Letters*. 2021 Jan 18;25(5):1720-4.

Iqbal N, Al-Dharrab SI, Muqaibel AH, Mesbah W, Stüber GL. Cross-layer design and analysis of wireless geophone networks utilizing TV white space. *IEEE Access*. 2020 Jun 26;8:118542-58.

Tanwar S, Gupta N, Iwendi C, Kumar K, Alenezi M. [Retracted] Next Generation IoT and Blockchain Integration. *Journal of Sensors*. 2022;2022(1):9077348.

Badshah A, Iwendi C, Jalal A, Hasan SS, Said G, Band SS, Chang A. Use of regional computing to minimize the social big data effects. *Computers & Industrial Engineering*. 2022 Sep 1;171:108433.

Iwendi C, Wang GG. Combined power generation and electricity storage device using deep learning and internet of things technologies. *Energy Reports*. 2022 Nov 1;8:5016-25.

Zhou J, Sun J, Zhang W, Lin Z. Multi-view underwater image enhancement method via embedded fusion mechanism. *Engineering applications of artificial intelligence*. 2023 May 1;121:105946.

Luo X, Zhang C, Bai L. A fixed clustering protocol based on random relay strategy for EHWSN. *Digital Communications and Networks*. 2023 Feb 1;9(1):90-100.

Ali J, Jhaveri RH, Alswailim M, Roh BH. ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. *Journal of King Saud University-Computer and Information Sciences*. 2023 Jun 1;35(6):101566.

Additional information may be obtained by writing directly to Hongfei Xiao, School of Information Engineering, Chuzhou Polytechnic, Chuzhou, Anhui 239000, China; Email: hxo02ao@163.com

AUTOBIOGRAPHICAL SKETCHES

Hongfei Xiao is at School of Information Engineering, Chuzhou Polytechnic, Chuzhou, Anhui 239000, China.

Wenwen Li is at School of media and design, Chuzhou Polytechnic, Chuzhou, Anhui 239000, China.

Shiqi Tang is at Information Ccenter of Minstry of Science and Technology, Beijing, 100862, China.

Manuscript revisions completed 30 November 2025.