Remote control and data analysis system based on Alibaba Cloud

Yuyi Zhai¹, Tao Jiang, Yunjia Liu*, Le Wang, Houyong Jiang

¹School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

*Corresponding author e-mail: smea2019@shu.edu.cn

Abstract. In order to eliminate the digital gap between equipment control and production information management, a smart factory remote control and data analysis system based on a cloud-fog hybrid computing model is proposed. Alibaba Cloud Server serves as the cloud computing layer to provide data storage and services, and the Internet of Things platform serves as the fog terminal. The computing layer completes data filtering and system early warning. The system consists of five parts: perception layer, gateway layer, platform layer, application layer and persistence layer. Each layer is designed and analyzed. The dead zone-adaptive swing door trending algorithm is used for data compression, and the system pressure test experiment and the data compression accuracy test experiment are completed. The experimental results show that this system improves the control of data in the production process, reduces the amount of data storage, and has high system compatibility to meet actual production needs.

1. Introduction

The tobacco industry is a highly confidential manufacturing industry. Production parameters must be strictly controlled and continuous production must be maintained. Tobacco companies urgently need to digitize production control. The goal of the informatization transformation of the production link is to significantly improve the existing data control by combining high-performance tools, low-cost sensors, and big data acquisition and analysis technologies. Combining the cloud platform with the Industrial Internet of Things, using the cloud-fog hybrid computing model to realize the management and control system of industrial sensing equipment, improving production efficiency, and simplifying system maintenance are the main development trends of smart factories in the future.

Meloni proposed and verified a cloud IoT-based architecture solution [1], which combines cloud computing capabilities and edge computing advantages, and uses virtualization technology to separate the processing of measurement data from the underlying physical devices. Usamentiaga proposed an architecture based on the IIoT paradigm [2] to provide solutions for rapid configuration, application deployment, and online service generation.

Rahmati proposed to train a neural network with the Internet of Things system as the core to complete the collection of leak point data, part of the data is used to detect the neural network characteristics of the fault detection system, and the other part of the data is used as a training set to continuously adjust the weight of the neural network [3]. Wan proposed a software-defined IIoT architecture, which will help flexible Industry 4.0 network resource management [4], giving the functions and characteristics of each component layer and the information exchange between various devices. In the background of Industry 4.0 a simplified evaluation test platform is shown below. Wu built a workshop equipment monitoring system based on interactive processing modules, data storage modules, and terminal display
modules to realize the rapid development and reconstruction of flexible configurable production lines [5].

These scholars' research on smart factories is more reflected in data processing and fault diagnosis, without too much consideration of the unified management of sensor equipment, timeliness of data transmission and data storage related issues, and no standardized management of workshop production resources.

2. System analysis
The off-site technical transformation of a tobacco factory in Shanghai has basically been completed, and some of the stem and expanded silk produced in the new factory need to be transferred to the old factory. The production process is shown in Figure 1. The existing transfer mode still uses cartons for manual packing and then transfers. There are problems such as high labor intensity, low packing efficiency, poor working environment, difficult management and control, etc. And it does not conform to the national industry 4.0 orientation[6]. This system can eliminate the digital gap between equipment control and production information management, and meets the needs of national scientific and technological development. The technical block diagram is shown in Figure 2.

2.1. System structure
This system is based on the cloud-fog hybrid computing architecture. Alibaba Cloud Server serves as the cloud computing layer to provide data storage and services [7], and the Internet of Things platform serves as the fog-side computing layer to complete data filtering and system early warning [8].

2.2. Perception layer
Tobacco is a flammable and moisture-prone item. The humidity of the workshop environment must be ≤80%, and the ambient temperature must be strictly controlled at 5 to 50 degrees. The temperature and humidity sensors are distributed in each area of the workshop. Tobacco production workshops need to be strictly fireproof, and real-time fire safety detection is carried out. The photoelectric smoke sensor can detect the smoke generated during the fire and is easy to install.

The production equipment is basically noise-free and runs smoothly during normal operation. The vibration displacement, speed, and acceleration are sensed by the vibration sensor to complete the characterization of the vibration phenomenon and reflect the vibration state of the motor during operation. The noise test can be used to analyze the operating status of the equipment, using high-precision sound measuring instruments and high-sensitivity condenser microphones to complete the equipment's noise signal collection. When the vibration and noise exceed the set limit, the alarm will be triggered.
2.3. **Gateway layer**

An IoT gateway is a device that allows traditional industrial equipment to use sensors or embedded controllers to transmit industrial production data to a remote server. The IoT gateway adopted by this system uses Modbus protocol for data transmission, and sends instructions to read the temperature and humidity value of the device address through the inquiry frame. The response frame shows that the number of valid bytes is 4, and the corresponding data can be obtained by converting hexadecimal to decimal and performing floating point shift.

2.4. **Platform layer**

The Alibaba Cloud IoT platform provides safe and reliable connection and communication functions for devices, connects a large number of devices to the platform, supports device data collection and upload in the cloud, and accesses cloud APIs. The device establishes a long connection with the IoT platform through the message queuing telemetry transport (MQTT) protocol, and reports data (publishing Topic and Payload through Publish) to the IoT platform. Configure the rule engine, write SQL to process the reported data, and configure forwarding rules to transfer the processed data to the Alibaba Cloud server through the advanced message queuing protocol (AMQP) consumer group. The server calls the API interface Pub based on the HTTPS protocol, sends instructions to the Topic, and sends the data to the IoT platform.

The MQTT network protocol can be implemented on severely limited device hardware and high-latency/limited bandwidth networks, making it possible to provide support for diverse application scenarios of IoT devices and services. Kafka is mainly used to process message queues in a large amount of data state[9], which can realize decoupling, asynchronous processing, and traffic peak cutting.

2.5. **Application layer**

The storage and processing capabilities of sensor nodes for industrial Internet of Things data collection are prioritized, and cloud computing can store for a long time and perform complex analysis in it. Traditional industrial monitoring systems generally use wired forms, which leads to the limitations of the entire system. Production personnel and management personnel can only perform on-site monitoring, and data between production equipment cannot be shared in real time, which increases production costs.

The application layer can access the database, and compare and analyze the collected data of the communication node with the early warning strategy table, and push alarm information to the user in time when the data is abnormal, and the user will make corresponding processing according to the alarm level. Industrial equipment data and operating status can be accessed in real time through browsers and mobile terminals, and high-privileged users can complete the cloud control of the equipment through instructions.

![Figure 3 Factory monitoring screen](image-url)
2.6. Persistence layer
This system uses the MySQL cluster structure to design the master-slave separation mode, and uses SQL to copy the operation log of the database in the operation of the master database and synchronize the auxiliary database, so that the data of the master database and the slave database are consistent, so as to realize the separation of write operations and read operations. The Redis sentinel process is used to monitor the working status of the Master master server in the Redis cluster. When the Master master server fails, it can switch between the Master and Slave servers to ensure the high availability (HA) of the system.

3. Adaptive Dead Zone- Swing Door Trending Data Compression Algorithm
In real-time detection of workshop data, an equal time interval data collection algorithm is generally used. According to the system setting, it is set to collect data once every 5s and upload. By calculating the data collection volume of a sensor a day, the file size is about 487.2k, which will generate approximately 16,704 pieces of data. By deploying more than 20 sensors for each device and about 30 sensors for the workshop environment, the data packet size generated every day is about 1.67Gb, and the data volume is 2,505,600 pieces of data. Among them, the data capacity occupied by vibration signals and noise signals will be greater.

The dead zone compression algorithm only processes when the data changes, and the data will not change during many time intervals in the industrial field, or there will be slight fluctuations, which will not cause interference to production. As shown in the figure, the blue dead zone in the Figure 5 shows that there is no obvious change in the data, and only the inflection point of the corresponding data needs to be recorded. When the data exceeds the dead zone, it will be stored as the inflection point of the data. The dead zone compression algorithm can filter out low-meaning data, optimize the network transmission quality, and enhance the robustness of the system.

The swing door trending algorithm performs high-quality compression on the data after the initial screening of the dead zone compression algorithm [10] to maintain the real-time trend of the data. As shown in Figure 4, determine the starting data point, the data compression accuracy, determine the next data point, and the third data point to construct a parallelogram with twice the compression accuracy as the side length, and do not record in the parallelogram area. Once a data point out of the area appears, record the previous point as a new initial data point, and continue processing until all data is compressed in this time interval.

![Program flow chart of adaptive swing door trending compression algorithm.](image)

![Schematic diagram of dead zone and swing door trending algorithm.](image)
This system uses the dead zone compression algorithm combined with the adaptive revolving door algorithm to compress industrial real-time data, continuously optimize the system, complete the adjustment of the compression accuracy, and obtain the largest possible compression ratio and the smallest possible error. Finally, the optimal compression accuracy of the system is determined.

4. System test

The use of sections to divide the text of the paper is optional and left as a decision for the author. Where the author wishes to divide the paper into sections the formatting shown in table 2 should be used.

4.1. Visit stress test

The system deployed by the cloud server adopts the B/S architecture and is accessed by users in the form of browser Web pages. When the user base of the system increases, it will inevitably lead to an increase in the number of visits per unit time. At this time, the system needs to have high reliability and strong stability. Considering that the maximum number of users of this system is about 800 people, the Apache Foundation's open source web stress testing tool ApacheBench is used. ApacheBench is a command line tool that has low requirements for the local machine that initiates the load. Use operation commands to create a large number of visits. Thread simulates a high-concurrency running state, and at the same time accesses a certain unified identification location address, and performs a stress test on the application server. Generally use the -c and -n parameters for testing. For example, abs.exe -c 1000 -n 3000 http://192.168.73.1:8080/shu_iot_cloud/login.jsp, where -n 3000 means the total number of requests, -c 1000 means 1000 users are accessing at the same time, the URL is the test Test entry for server operation.

For example, Table x is the relevant configuration of the cloud server of the system, and the experimental operation interface shown in Figure 7 Failed request is 0, which can be judged that the experiments in this group are all operating normally and are highly effective. Through a large number of concurrent access tests on the login interface, the experimental results shown in Figure 6 are obtained. The concurrent threads are set from 20, 50 to 1000. After the experiment, the access requests per second are between 485 and 520, which fully meets the actual production. Demand, and the server version of the final configuration has better performance.

4.2. Data compression test

This system compresses data by combining dead zone compression and revolving door compression algorithm, compresses the data set by setting different initial compression accuracy, and then performs decompression curve fitting on the compressed data set. From Figure 8, 9, 10 it can be concluded that when the compression accuracy ΔE is 0.01, 0.03, 0.05, a good data curve can be maintained. When the compression accuracy is greater than 0.05, the data will have obvious errors, and when the compression accuracy is greater than 0.1, the original trend is basically completely lost. According to the experimental results, when the compression accuracy ΔE is around 0.05, the storage space can be saved by more than 50% and the loss of data accuracy is small. Considering that the temperature is tested in this experiment,
the compression accuracy setting of other acquisition parameters needs to be passed Obtained by experimental optimization.

5. Conclusion

This paper proposes a smart factory remote control and data analysis system based on a cloud and fog hybrid computing model, and tests the system performance, compression accuracy, and data accuracy to meet the required functional requirements. The perception layer uses the Modbus protocol to connect with the IoT gateway. The IoT gateway uploads Json format data to the Alibaba Cloud IoT platform through the MQTT protocol. The IoT platform serves as a fog node to complete preliminary fog calculations for data filtering and system warnings. The data is uploaded to the cloud through the Kafka message queue through the AMQP protocol for data compression, data storage and data analysis, and PC and mobile device access are provided to control the system. This system improves the management and control of production link data, simplifies system maintenance, reduces data storage capacity, improves data usage efficiency, and has high system compatibility and can be transplanted to other production links.

References

[1] Meloni A, Pegoraro P A, Atzori L, Benigni A and Sulis S. Cloud-based IoT solution for State Estimation in Smart Grids: exploiting virtualization and edge-intelligence technologies [J]. Computer Networks, 2017:S1389128617303912.
[2] Usamentiaga R, Fernandez M A, Villan A F and Carus J L. Temperature Monitoring for Electrical Substations Using Infrared Thermography: Architecture for Industrial Internet of Things[J]. IEEE transactions on industrial informatics, 2018.
[3] Rahmati M, Yazdizadeh H, Yazdizadeh A. Leakage detection in a gas pipeline using artificial neural networks based on wireless sensor network and Internet of Things[C]// 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, 2017.
[4] Wan J, Tang S, Shu Z, Di L, Wang S and Imran M. Software-Defined Industrial Internet of Things in the Context of Industry 4.0[J]. IEEE Sensors Journal, 2016, 16(20):7373-7380.
[5] Ar A, Ab A, Mb A and Esm B. The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions - ScienceDirect[J]. Procedia Engineering, 2014, 69:1184-1190.
[6] Malloy L E, Gingerich J, MD Olson and Atkins D L. Remote Monitoring of Cardiovascular Implantable Devices in the Pediatric Population Improves Detection of Adverse Events[J]. Pediatric Cardiology, 2014, 35(2):301-306.
[7] Yu S. Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data[J]. IEEE Access, 2017, 4:2751-2763.
[8] Priyadharshini B. Data integrity in cloud storage. IEEE, 2012.
[9] Puschmann T, Alt R. Enterprise Application Integration - The Case of the Robert Bosch Group [J]. IEEE Computer Society, 2001.
[10] Yang G, Li Y. Analysis of industrial real-time data compression algorithm[J]. Automation and Instrumentation,2016(08):214-215.