Design and Implementation of Real Time Monitoring System of Textile Equipment Based on MQTT

Shaoheng Han1, Yiran Chen1, Songlin Ye1
1Shanghai Electric Group Central Academe, No.960 Zhongxing Road, Shanghai, PRC.
E-mail: hanshh@shanghai-electric.com

Abstract. At present, with the progress of network information technology, the idea of intelligent factory and intelligent manufacturing is imperative. It is imperative to realize intellectualization and informatization in traditional manufacturing plants. According to the intelligentized demand of textile equipment, the real-time monitoring of the spindle status of the textile machine is designed and realized, and the functions of the real-time statistics of the operation efficiency, the number of broken ends, the spindle status of the textile machine, the statistical report and so on are made in real time. It has achieved remote and efficient production management and fault detection. It effectively reduces the cost of manual operation and maintenance, and improves the intelligence level of textile mills.

1. Introduction
At present, with the rapid development of the textile industry, a variety of new textile machines [1] have been introduced. The function of textile machine is constantly upgrading, relying on the traditional manual control, can not meet the growing needs of textile mill scale. The establishment of a global monitoring system can greatly facilitate the management of large-scale equipment [2]. At the same time, it can save data for a period of time, generate reports regularly, and calculate the production efficiency. It can eliminate faults in time, reduce operation and maintenance costs, realize real-time remote monitoring, and improve the intelligent degree of textile workshop.

2. Overall system architecture
As shown in Figure 1, the system is divided into three layers, including data acquisition layer, data storage layer and web system presentation layer. The textile machine sends data to the remote server through tcp/ip protocol, and the data parsing program is deployed in the remote server to complete the data analysis and warehousing. Web system reads data from local database in real time, calculates and shows.
Specific steps:
Step 1: the local machine of Hunan Textile Mill collects the machine data and sends it to the remote computer receiver through socket;
Step 2: the remote computer receives the data and writes it to the memory queue;
Step 3: the remote end obtains data from the memory queue in real time and stores it into mysql;
Step 4: deploy the web monitoring system on the remote end to obtain and display the data from the database;

3. Data acquisition module
The data collection side uses mqtt [3] protocol, publish / subscribe mode for data collection. Mqtt (message queuing telemetry transport) is a standardized publish / subscribe message transmission protocol, which was designed in 1999, and was originally used on satellites and other objects. It is a very lightweight protocol. Due to the low bandwidth requirement, it has become an ideal choice for M2M communication or Internet of things applications. Now it has become one of the most common protocols in this kind of scenario.

Publish / subscribe, also known as pub sub mode, is the core of mqtt. In addition to the publishers and subscribers based on the same message broker, there are some other nodes distributed around the message broker in a star topology. This model is very different from the standard client / server, which seems strange at first, but the decoupling capability it provides has great advantages in many cases.
Clients can publish or subscribe to specific topics (topics, some similar to information topics) and decide who will receive information according to the message broker that uses them. The theme of mqtt has a specific syntax, using slashes (/) as separators. The overall structure is hierarchical, very similar to the path format in URL. Therefore, photoelectric sensors in the machine may be published to topics like "sensors / machine / spindle".

Let's take an example: imagine a network connecting spindle sensors in the whole workshop to provide condition monitoring service. The middleware keeps all of these messages connected to the current agent every five minutes. Based on their own location, they publish information to specific topics in the following format: sensors / state / {machine} / {spindle}. Then a sensor in a workshop will publish a piece of information containing the current running status to "sensors / state / Center / monitor".

The monitoring service needs to ensure that the data of the historical status database is up-to-date. Therefore, the database service subscribing to the mqtt topic is created, and the database service will give a prompt when it receives the latest status information. However, there is a problem: the database service needs to know all the status sensors in the whole factory, and it will be very complicated to subscribe each sensor to a separate topic. Fortunately, mqtt has a corresponding solution: wild cards.

There are two wildcards available in mqtt, which are + and #. + means to match any topic in a single level, and #, means to match any number of levels. Therefore, there may be a service subscribing to sensors / state / #, which can receive status readings from any sensor in the whole plant.

However, if users want to use these data in their own monitoring service, they can only subscribe to sensors / temperature / machine / #, and only accept the sensor readings of the specified machine. If a service wants to receive all types of sensor data from a particular spindle, it can use a format similar to this: sensors / + / machine. As you can see, this is an excellent modular system. Adding new sensors and databases is just a piece of cake. Mqtt message broker can be highly parallelized and event driven, so that a single message broker can easily expand to the level of processing tens of thousands of messages per second.

Quality of service (QoS). Mqtt is designed to work well in unreliable networks. It provides three levels of QoS for different scenarios and allows clients to specify the reliability level they want.

QoS level 0: at least once, this is the simplest level, without client confirmation, and its reliability is consistent with the TCP / IP of the basic network layer.
QoS level 1: at least once, it is possible to repeatedly ensure that the information is sent to the client at least once, but it can also be sent multiple times; when receiving data packets, the client needs to return an acknowledgement message (ACK packet). This approach is often used to deliver information that ensures delivery, but developers must ensure that their systems can handle duplicate packets.

QoS level 2: only once, ensuring that messages arrive only once, which is the least common level of quality of service, ensuring that messages are sent only once. This method needs to exchange four packets, and it also reduces the performance of message broker. Due to the relative complexity, this level is usually ignored in mqtt implementations. Make sure to check for this before selecting a database or message broker.

![Diagram](image.jpg)

Figure 3. Service quality level partition in mqtt

The protocol provides a detection method, and uses keepalive mechanism to find problems when the client is disconnected abnormally. Therefore, when the client runs out of power, crashes or the network is disconnected, the message broker will take corresponding measures. The client will send the "dying will" (LWT) information to the message broker at any point. When the message broker detects that the client is offline (the connection is not closed), it will send the LWT information saved on a specific topic to let other clients know that the node has been offline unexpectedly.

Security, the security of mqtt is a considerable topic, but in this paper, only two main security functions are involved: authentication and encryption.

Authentication is realized by sending the user name and password in the mqtt connection package, which is supported by almost all message agents and clients. But because information is too easy to be intercepted, in order to avoid it, we should use TLS as much as possible.

The protocol itself does not provide encryption function, but because mqtt runs on the upper layer of TCP, we can easily use TLS to provide encrypted connection. But it does increase the computational complexity of sending and receiving information, which will not only cause problems in constraint system, but also affect the performance of message broker.

Mqtt is an ideal protocol, and its application in the communication between Internet of things and M2M is unlimited. If you need a lightweight messaging system, it's a good choice, and it's likely to catch on in the next few years.

4. Web system display module

The monitoring system adopts b/s architecture as a whole, which is mainly divided into real-time data display, report statistics, authority management and real-time alarm. Real time data display the data collected through the front end and feedback to the front end in real time. Report statistics: calculate the report data and form a report by week and month. Authority management can hide information according to users of different companies and regions. Set the alarm threshold, alarm and record the test points that exceed the threshold.
5. Existing problems
With the increasing amount of data, there are higher requirements for concurrency. At the same time, due to the continuous accumulation of data, the capacity of MySQL database has gradually reached the bottleneck, resulting in the decline of data storage and reading performance. Existing problems: simulate one machine to collect the running status information of 48 ingots once per second, the required storage space: 138byte * 60 * 60 * 24 / 1024 * 1024 = 545.8m; actual one machine to run the required storage space: 1200 / 48 * 545.8 = 13gb; 10 sets of 10000 ingots need 130gb per day; excessive data transmission will affect the read-write performance of MySQL, resulting in data loss.

6. Conclusion
This paper mainly refers to the overall architecture of textile mill data acquisition system [5]. Based on mqtt, the real-time status information of the plant equipment is obtained, and the real-time monitoring system realizes the real-time online display and analysis of the operating parameters of the textile machine. At the same time, it ensures the storage of short-term data, reduces the cost of manual operation, facilitates the enterprise to realize the global equipment information monitoring, and meets the requirements of application scenarios.

Reference
[1] Lizijun, Wang Quan. Design of new type network air-jet loom control system [J]. Modern textile technology, 2018,26 (1): 73-76.
[2] He Jiafeng, fan Yanbin. Hardware design and analysis of weft insertion control system of air jet loom based on STM32 [J]. Journal of Qingdao University (Engineering Technology Edition), 2012 (3): 74-78.
[3] Ren Heng, Ma Yue, Yang Haibo, Jia Zhengfeng. Message push server based on mqtt protocol [J]. Computer system applications, 2014,23 (3): 77-82.
[4] Qiu Haifei. Comprehensive technology progress and research direction [J]. Cotton textile technology, 2016,44 (5): 80-84.
[5] Zhan Yougang. Creo3.0 mechanical design course [M]. Beijing: China Machine Press, 2014.