Application of Web Socket in Dynamic Face Recognition System

Diyu Chen

ABSTRACT

With the executing of the project to build a safe city, a smart city, a large number of sky-net cameras are being used. The efficiency of the public security police to solve cases is greatly improved by monitoring and viewing the camera. At the same time, with the development of the face recognition intelligent analysis algorithm, the dynamic face recognition system has been widely applied in practical production. Though analysis algorithm has achieved a good effect, the results of the analysis need to be displayed to users in real-time, so that it can manifest its value. The delay of the data will make public security officers miss opportunities, which will go against criminal detection.

Based on the Web Socket protocol and taking the dynamic face recognition system as an example, this paper describes the application of Web Socket protocol in the dynamic face recognition system. Through the applying of Web Socket in the system, the data is transmitted to the client in real time which makes the real-time communication become true and which maximizes the value of the information.

INTRODUCTION

Web Socket protocol is a new network protocol based on TCP. It implements the full-duplex communication between the browser and the server - allowing the server to actively send message to the client[1]. The Web Socket protocol supports full duplex communication between the client (which runs the untrusted code under controlled environment) and the remote host (which is chosen to join the code). The
security model used for this is a common based-on-original security model for Web browsers. The protocol includes an open handshake and the following message frames on the TCP layer. The goal of this technology is to provide a communication mechanism for the application which is based on the browser, need a two-way communication with the server (the server cannot rely on opening multiple HTTP connections.

APPLICATION OF WEB SOCKET

As is known to all, communication process of Web applications is usually like this: the client sends a request through the browser[2], and after the server receives the request, it processes and returns results to the client, and then the client browser will display the message. This mechanism for those applications whose information change is not particularly frequent can be a good support, but for those highly requiring real-time and huge amounts of concurrent, it appears stretched, especially under the current trend of the vigorous development of mobile Internet industry, during which high concurrency and real-time response are problems which Web users will usually meet, such as the real-time information for financial securities, location acquisition of Web navigation application, real-time news push of social networks, etc.

When dealing with such business scenarios, the traditional request-response pattern of Web development is usually live communication solutions. Such as common polling solution, whose principle is simple and easy to understand. With this solution, the client continually sends a request to the server at a certain time interval to keep the synchronization of data between the client and the server. However the shortage is also clear: when the client sends a request to the server at a fixed frequency, data in server-side may not be updated, which will bring a lot of unnecessary request to cause waste of bandwidth and low efficiency.

In dealing with high concurrent and real-time needs, the traditional Web model will encounter insurmountable bottlenecks, so it is necessary to adopt a highly efficient, energy-saving and bidirectional communication mechanism to ensure real-time data transmission. In this context, WebSocket (named Web TCP) based on HTML5 specification came into being. Early HTML5 did not form a unified industry norms, each browser and application server vendors have different but similar implementations, such as IBM's MQTT, Comet's source -opening framework. Until 2014, HTML5 finally settled to become the actual standard specification, and the application server and browser vendors gradually began to unify and also achieved WebSocket protocol in JavaEE7. At this point both the client and server-side WebSocket are complete.

Web Socket Mechanism

WebSocket is a new protocol under HTML5. It implements full-duplex communication between the browser and the server, which can save server resources
and bandwidth better and achieve real-time communication. It is the same as HTTP to transfer data through the established TCP connection, but the biggest difference between HTTP is that WebSocket is a new protocol under HTML5. It implements full-duplex communication between the browser and the server, which can save server resources and bandwidth better and achieve real-time communication. It is the same as HTTP to transfer data through the established TCP connection, but the biggest differences between it and HTTP are that, compared to the traditional HTTP which requires the client and the server to establish a connection every request and answer[3], WebSocket is similar to the Socket which uses TCP long connection communication mode. Once the WebSocket connection is established, subsequent data is transmitted as a frame sequence. The client and server do not need to re-initiate the connection request before the client breaks the WebSocket connection or the server-side breaks the connection. In the situation where there are massive concurrency and interaction between the client and the server causes heavy load traffic, WebSocket greatly saves the consumption of network bandwidth resources, which means that it has obvious performance advantages, and the client to send and receive messages is launched on the same persistent connection, so its real-time advantage is also obvious.

**The Advantages of WebSocket over HTTP**

WebSocket is an actual full-duplex mode, with which the client and the server side is completely equal after the establishment is done, and both sides can actively request each other. While HTTP long connections based on HTTP is the traditional mode with which client sends request to the server.

In HTTP long connection, the server and the client also exchange a large number of HTTP header besides the real data part when exchange data, which makes the efficiency very low. With Websocket, there is no need to send HTTP header to exchange data once the first TCP connection is established through the request, which obviously differs than the original HTTP protocol. So it need to upgrade the server and the client then it can be achieved (mainstream browsers already support HTML5). There are also multiplexing, different URLs can reuse the same WebSocket connection and other functions. These cannot be done with HTTP long connections.

**Application of WebSocket in Dynamic Face Recognition System**

WebSocket makes Web application managed by the browser can stay connected to the web endpoint, which will also reduce cost (such as the heaviness of the server, memory and resource consumption) to a minimum. The actual effect is that the data and notifications can be sent and received without delay between the browser and the Web server, and without extra requests[4]. Dynamic face recognition system applies this feature of WebSocket, making the alarm and capture data can be efficiently pushed to the browser, so that the police can timely access to information
of suspects during the criminal investigation, and this provides favorable conditions for arresting suspects.

**Implementation Steps**

1. To build Java Message Service (JMS) middleware, in which JMS Provider adopts Apollo deployment;
2. The client implements full-duplex communication with the JMS Provider message with the WebSocket protocol.
3. To establish analysis tasks, and then push analysis results to the JMS middleware through the interface;
4. JMS monitors and receives data and stores it in the database;
5. JMS middleware takes the initiative to send a message to the Web browser, and receives and displays the capture results and the comparison results in real-time.

For analysis tasks, it is necessary to call the related interface according to the service agreement when sending messages to JMS middleware. After the establishment of WebSocket connection is done, the video data is analyzed by the algorithm analysis of server, and sent to JMS message middleware by calling interface, and then JMS message middleware directly push the data to the client, and then the full process is complete.

**CONCLUSIONS**

WebSocket is widely used in a variety of applications due to its unique advantages, such as social subscriptions, stock fund quotes, sports updates, etc. This paper focuses on the WebSocket mechanism and its advantages, and combined with a company's dynamic face recognition system to describe the application of WebSocket in this system, which makes the system achieve the display of capture data and the alarm data in real-time. But WebSocket also has its limitations, for example it cannot be developed for those browsers do not support WebSocket, so this needs further research and optimization.

**REFERENCES**