Design of Instant Messaging System of Multi-language E-commerce Platform

Heng Yang*, Xinyi Chen, Jiajia Li, Yaru Cao
Northwest University for Nationalities, National languages Information Technology research institute, Lanzhou city, Gansu province, China
*Corresponding author

Abstract. This paper aims at researching the message system in the instant messaging system based on the multi-language e-commerce platform in order to design the instant messaging system in multi-language environment and exhibit the national characteristics based information as well as applying national languages to e-commerce. In order to develop beautiful and friendly system interface for the front end of the message system and reduce the development cost, the mature jQuery framework is adopted in this paper. The high-performance server Tomcat is adopted at the back end to process user requests, and MySQL database is adopted for data storage to persistently store user data, and meanwhile Oracle database is adopted as the message buffer for system optimization. Moreover, AJAX technology is adopted for the client to actively pull the newest data from the server at the specified time. In practical application, the system has strong reliability, good expansibility, short response time, high system throughput capacity and high user concurrency.

1 Introduction*
Along with the rapid development of the Internet, the network application is diversified and IM (Instant Messaging) gradually becomes the hotspot of Internet application. IM refers to a method for transmitting information to users in a real-time manner. Due to instantaneity, rapidness and convenience, IM becomes an important tool for human communication in network environment to gradually replace emails.

Along with the promotion of IM software, relevant research on IM software based activities is carried out successively, but some disadvantages and problems also exist in practical application. Therefore, existing problems were analyzed in this paper in order to propose Tomcat based instant messaging system and research this system and relevant contents.

2 Research Content
The instant messaging system as an online user communication tool can be conveniently updated & upgraded and integrated with other systems in order to maximally improve user satisfaction and user experience. The users concerned mainly include common user and administrator. Specifically, the administrator can effectively manage the system and common user; the most common function for the common users is to realize instant messaging and online communication at any time, transfer files during communication, inquire communication logs, etc.

*This research is supported by National Science-technology Support Plan Projects (NO. 2015BAD29B01).
2.1 System objective
According to demand analysis, the instant messaging system should achieve following objectives:

1. In order to avoid the troubles for software downloading, installation and upgrading, B/S architecture should be adopted.
2. Besides low time delay and messaging smoothness, the system should be available for the online communication of millions of users.
3. The system should have good expansibility. In case of business change, the system should adapt to new business demands and should be convenient for expansion.
4. The system should have good reliability and should ensure that the messages are not lost or can be recovered under special conditions, and all messages should be persistently stored in the database to ensure ordered messages.
5. The instant messaging system interface should be simple, intuitional and convenient in order to improve user satisfaction.

2.2 System architecture
In order to improve system reliability, the dual-computer hot standby mode should be adopted for all servers. Oracle memory database is adopted as user message buffer, so the messages transmitted and accepted in user communication process can be directly read from the memory and there is no need to read and write in the magnetic disk for each time, thus improving system throughput capacity and user concurrency. In order to avoid user message loss caused by any special reason, we particularly adopt one module to back up the data into MySQL database. The system architecture is as shown in Fig.1.

![System Architecture Diagram](image)

2.3 System implementation
3.1 Function implementation
According to above analysis, we can divide the system into the following several modules and the specific functions thereof are as follows:

1. Message transmission: users transmit messages to the server; the message transmission includes common transmission and group message transmission.
2. Message receiving: users receive messages from the server; message receiving includes common receiving and group message receiving.
3. Message transmission status: when users transmit messages to the server, the message system will monitor the message transmission process and timely indicate the users about the status of the present message, including un-transmitted message, transmitted message, received message, read message, etc.
4. Personal information completion and modification: a registered user can complete or modify her/his head portrait, nickname, gender and other basic information through settings.
5. User addition and deletion: for convenient management, the users can optionally add or delete friends.
(6) Communication log inquiry: the users can inquire the history of the chats with other users.
(7) Grouping: the users can create a group chat room to realize many-to-many chat.

The functional framework of the system is as shown in Fig. 2.

![Functional Framework of System](image)

---

Among above function modules, we take the most complicated module ---- Message Receiving as an example. The client processing flow thereof is as shown in Fig. 3.

![Message Receiving Flow Chart](image)

We make AJAX transmit a request to the server at the specified time; then, HTTP pull technology is adopted to know whether there is any message transmitted to the user. After inquiring in the database, the server sends the result back to the client. If there is a new message, the client will send the request immediately, and the server will inquire in the database again to acquire the message needed by the client and transmit the message thereto. Therefore, the client needs to frequently transmit requests to the server, and for each request, the server needs to inquire in the database to obtain the desired result. For the instant messaging system, due to the large number of online people, the server needs to frequently visit the database and the IO workload of the server is also increased, so the IO demand of the system becomes particularly important. For realizing the instant messaging system for the online communication of millions of people, IO optimization is a critical step.

3.2 System optimization

According to above analysis, when the instant messaging system has large user quantity, if the server needs to visit MySQL database for each message transmitted or received by the user at each time, the IO workload of the system will become larger and larger, thus inevitably increasing system time delay and reducing system throughput capacity as well as influencing user experience.

In order to solve above problems, we prepare an effective scheme. Specifically, we add a “message buffer” in the system for putting the newly received messages; after a new message is read by the user, this message will be stored in MySQL database and will be also deleted in the “message buffer”. In other words, the new messages are stored in the memory, and the speed for reading in the memory is obviously faster than that for reading in the magnetic disk. The persistent operation of the messages is asynchronous with message transmission & receiving, namely: data writing in MySQL will not increase the time delay of message transmission & receiving.
However, it is difficult to realize the “message buffer”, and we should consider both data effectiveness and program run performance. Therefore, we add an Oracle database between Tomcat server and MySQL database and take it as the “message buffer” to improve system performance while not increasing system complexity.

4 Conclusion

According to the demand of the instant messaging software for the message system, this paper aims at researching the message system in the instant messaging system based on multi-language e-commerce platform. Specifically, the high-performance server Tomcat is adopted to handle user operations; MySQL database and Oracle database are adopted for massive message caching to reduce system time delay and significantly improve system throughput capacity and user concurrency. Moreover, many common design modes and network application design modes are reasonably used in this paper.

Additionally, HTTP pull mode is adopted in this system for the communication between the client and the server, but this mode is insufficient to ensure data timeliness and effectiveness. In subsequent work, we also need to optimize and improve data timeliness and effectiveness, and adopt such technologies as HTTP stream and asynchronous polling technology to further optimize the system and improve system performance.

Acknowledgement

This research is supported by the Fundamental Research Funds for the Central Universities (NO. Yxm2016131).

References

[1] Lin Yan. Design of Instant Messaging System in LAN [J]. China New Telecommunications, 2013 (12): 8-13.
[2] Chen Hang. Design of Multi-function Network Instant Messaging System [J]. Network Security Technology and Application, 2014(11): 53-56.
[3] Pan Jinxue. Design of Enterprise Instant Messaging System Based on JMS [J]. China New Telecommunications, 2014 (15): 107-108.
[4] Liu Peihe, Niu Xiaolei, Du Peng, He Wencai, Zhang Yuanyuan. Security Analysis and Design of Enterprise Instant Messaging System [J]. Network Security Technology and Application, 2007 (2): 66-73.
[5] Zhang Peng. Application of P2P Technology in Instant Messaging and Security Strategy [J]. Information & Communications, 2015(01): 178-179.
[6] Feng Bo, Ying Qun. Object-oriented Java Network Programming [M]. Beijing: Tsinghua University Press, 2003.
[7] Yin Min. Java Language and Object-oriented Programming [M]. Beijing: Tsinghua University Press, 2000.
[8] Written by Bruce Eckel, translated by Chen Haopeng. Thinking in Java [M]. Beijing: Posts and Telecom Press, 2004