

Design and implementation of online automatic judging system

Haohui Liang¹, Chaojie Chen², Xiuyu Zhong³ and Yuefeng Chen*

^{1,2,*}Math and Computer College, GuangDong Ocean University, Zhanjiang 524088, China

³School of Computer Science, Jiaying University, Meizhou 514015, China

Abstract. For lower efficiency and poorer reliability in programming training and competition by currently artificial judgment, design an Online Automatic Judging (referred to as OAJ) System. The OAJ system including the sandbox judging side and Web side, realizes functions of automatically compiling and running the tested codes, and generating evaluation scores and corresponding reports. To prevent malicious codes from damaging system, the OAJ system utilizes sandbox, ensuring the safety of the system. The OAJ system uses thread pools to achieve parallel test, and adopt database optimization mechanism, such as horizontal split table, to improve the system performance and resources utilization rate. The test results show that the system has high performance, high reliability, high stability and excellent extensibility.

1 Introduction

With the development of computer software applications, software developers need more high-quality. Learners can effectively improve their programming skills from participating in a large number of contests and programming training. However the traditional artificial judge brings more labour costs and the lower accuracy of judge. The Online Automatic Judging System(referred to as OAJ system), which can be applied to various algorithm programming competition and training, can carry out and evaluate of JAVA code, automatically judging the correctness, and counting the efficiency indicators, such as running time and memory consumption.

In [1], a specific online judge for automata is designed. The paper [2] discusses some advantages when ACM contest mode is brought in programming language and algorithm design courses. In paper [3] two new evaluation methods based on the requirement of the “Data Structure” course is introduced. The paper [4] introduces the URI online judge tool to enhance traditional algorithms classes.

2 Architecture Design

2.1 Global design

OAJ system is based on B/S architecture, including two service part: security sandbox environment and Web part. When the system becomes more complex, single application would tend to worse scalability, harder resource optimization, higher maintenance cost etc. Thus the Micro Service Architecture (MSA) is put forward, supporting more flexible, lightweight, loosely coupled architecture design^[5]. The OAJ system is designed with MSA, which provides asynchronous upgrading between the sandbox judging side and Web side, and realizes the flexible deployment and expansion. The OAJ system architecture is shown in Figure 1.



2.2 WEB side design

The Web side is developed with the rapid development framework: Spring + Spring MVC + Mybatis (SSM), including four parts: the control layer, business logic layer, data persistence layer and basic service layer.

The Web side provides a visual interface for the general user and administrator, and records the operation information into the corresponding database; the Web side is also responsible for communicating between each sandbox, and processing register, login and other business logic operations (such as running the midnight tasks at night setting up by the administrator in advance). In order to enhance retrieval performance of the database, OAJ system has to horizontally split the table, for example, the table recording submitted codes which is rapid growth when the number of users increase. Thus the framework should had high flexibility and custom in operating the database, and allow the system to make sure the name of the data table until running. Comparing with the other popular ORM framework (such as Hibernate), Mybatis is an excellent persistence layer framework which has the ability of flexible customization and more meticulous optimizing to the query fields in SQL.

2.3 Security sandbox design

The OAJ system often encounters security threats from submitting malicious code, which may cause server failure, crash or hard disk blocking^[6]. Java provides a sandbox model (Sandbox Architecture) as security mechanism^[7]. When design the OAJ system, to ensure the safe operation of the codes, users must be permitted to submit codes which can only access limited resources within the sandbox provided. Therefore in order to achieve efficient and reliable security sandbox, the model based on the sandbox provided by JAVA was built, avoiding the user submitted malicious to damage the system. The security sandbox side is separated from the WEB side, interacting through an interaction protocol and standard procedures. The main task of the security sandbox judge the request from the web side, compile and run programs submitted by the users, then output the running result according to the preset standard input file, and then package and sent back to the web side. When the system needs to extend, only a new security sandbox should be rebuilt. The system has good scalability, when the businesses grow in the future, it may be convenient to deploy the security sandbox onto the other server, so as to improve the system performance. The expansion requires only modifying a small amount of codes to improve the stability of the network.

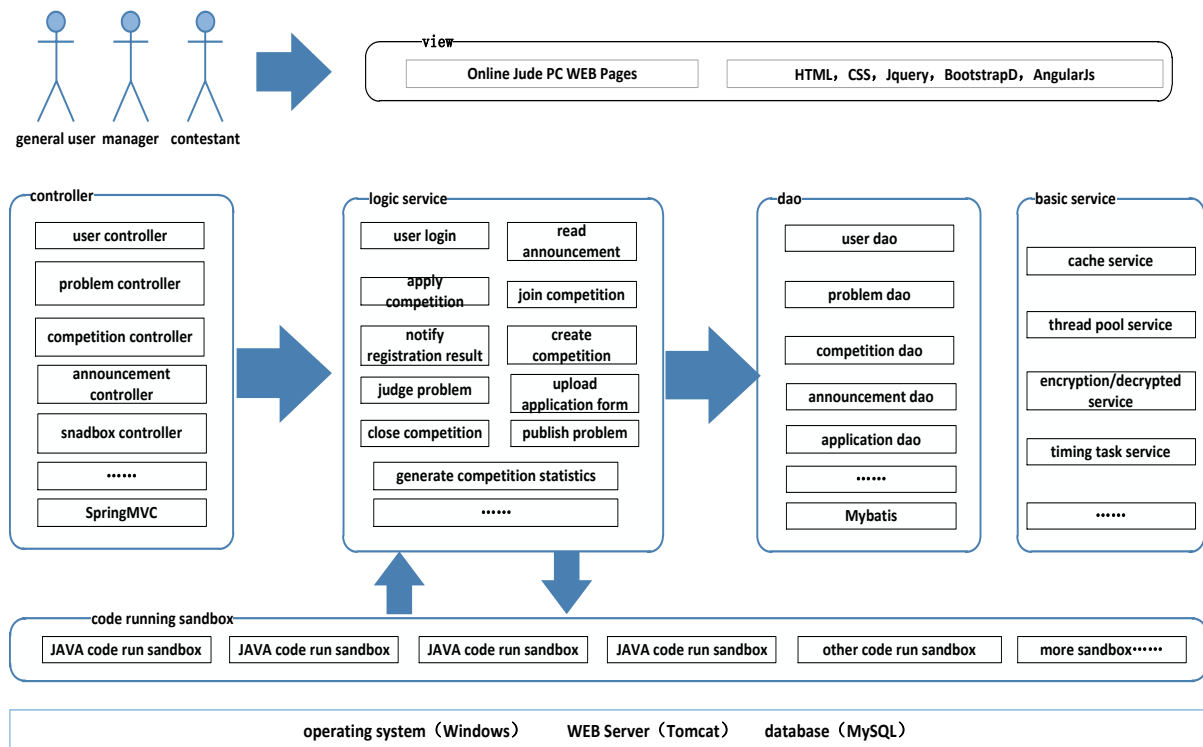


Figure 1: OAJ system architecture

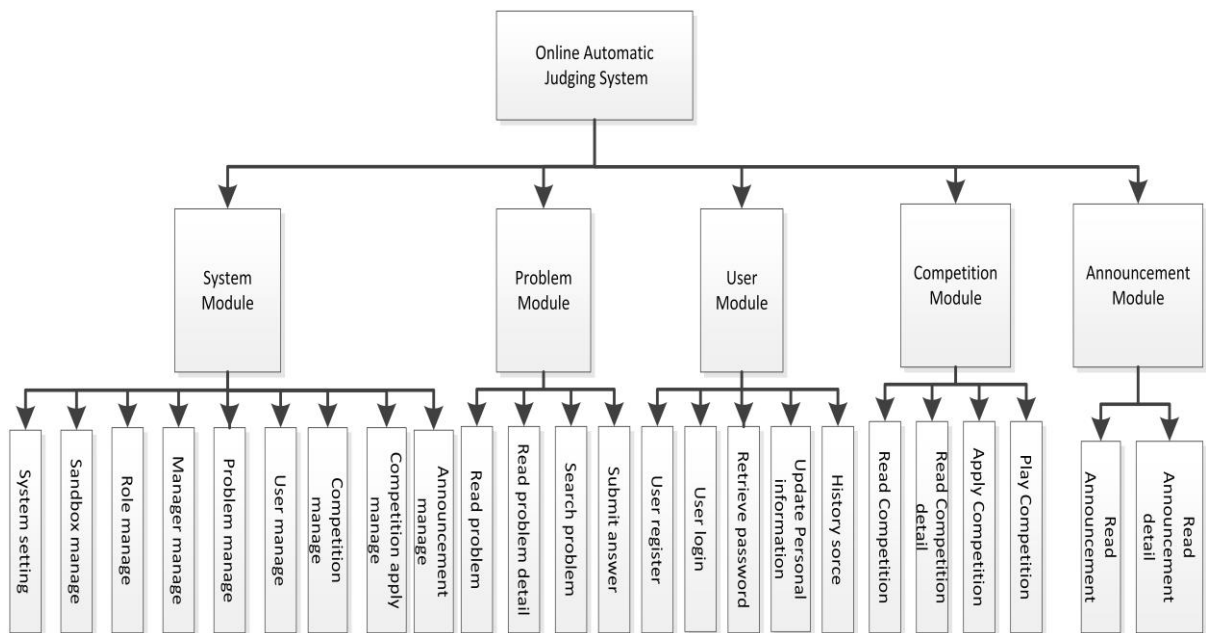


Figure 2: The hierarchical structure diagram

3 Software Design

3.1 Functional structure design

The OAJ system provides functions of question management, user management, system management for the administrator, and provides functions of joining competition and checking the score for the

user. The system has five modules, including question module, competition module, user module, announcement module and system module. The hierarchical structure diagram is shown in Figure 2

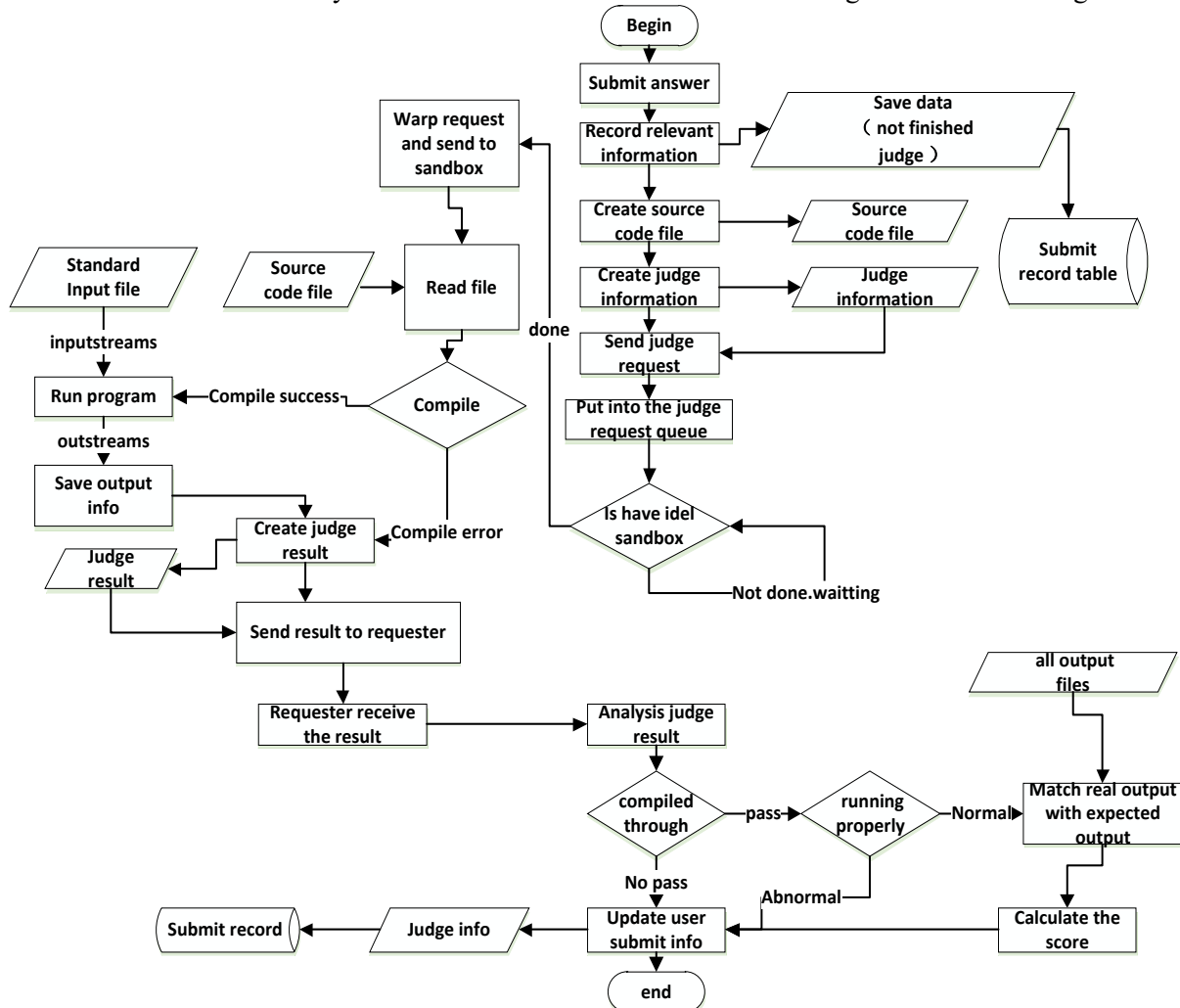


Figure 3: The procedure of judging process

3.2 The main business process design

Take the core business process, judging process, for instance, the procedure is: the web side collects submitted source codes from users, then packs and sends the request to the sandbox; The sandbox compiles and runs the received codes, then output the result according to the prepared standard input, and then packs results back to the Web side; finally the Web side matches the return results with the expected output, and calculates the detail scores. The procedure of judging process is shown in Figure 3.

4 Key Technologies

4.1 Parallel judgment

Each JAVA code has only one static main method after compiled, which is the running entrance. When making judgment, OAJ system calls the main method to obtain the output of the program through the reflection mechanism of JAVA. For normally each program must have multiple test cases and each test case is independent, the threads of judging program design with dynamic Cached Thread

Pool of JAVA, for each dynamic Cached Thread can work independently, without mutual interference and deadlock. Therefore, the parallel judgment can improve the system efficiency. On the other hand, for the Cached Thread Pool, the system will call back and release memory to reduce waste of the memory resource when the thread is idle for more than 60 seconds.

4.2 Split input/output stream

When running the test cases, most programs require the system input stream, but the reaching time and needing time are uncertain. Thus if we set the system input stream as a file stream by means of System.setIn() method, there will be the problem of concurrency, which may lead to a certain thread may not get the input stream, or get the input stream from the other threads. That may take muddle and affect the validity of the judge, so as to the output stream. In a multi-thread environment, if multiple test cases output their results to the same output stream, then the output information will become chaos, thus affecting the correctness of the judge.

In order to correctly splitting the input or output stream, OAJ system set the standard input/output stream (inheritance Inputstream/Outputstream) agent, which will bound each thread with the corresponding input or output stream. When a thread needs to obtain system standard input or output the result, it will match the corresponding real input or output stream according to the current thread ID. The principle of splitting input/output stream is shown in figure 4.

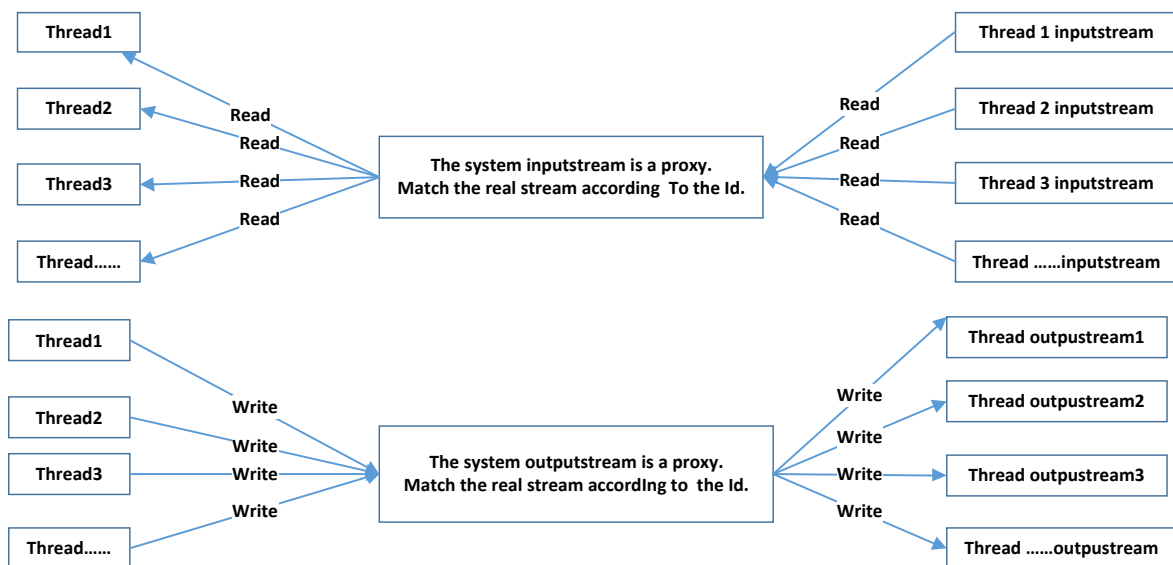


Figure 4: The principle of splitting input/output stream

4.3 Signal interaction mechanism

The sandbox side and Web side should communicate according to the defined signals in the network. There would be a complete set of communication signal and rules, and the necessary test process, to form a sandbox specification and abstraction. The signal interaction mechanism provides information interaction between Web side and sandbox side, and informs the sandbox side through signals, such as the web side want to make sure what services are provided from the sandbox side, what kind of function to realize, and how to use the sending data.

The major signals in the OAJ system are judging requests, viewing the sandbox status, closing sandbox etc. For example when the web side sends end judge request to the sandbox, it will package the Judge signal to the request; after the sandbox side receiving the request, it analyzes that the request contains a Judge signal, the sandbox will interpret the remaining data as the required information for the judgment process, and then use the information as the source code to compile and run, and then encapsulate the operation results, as a request result, return to the web side.

4.4 Message box mechanism

For sometimes communications between the Web side and the sandbox side through the network are slow, so it must not be synchronized but asynchronous when the Web side waiting for the sandbox side to return the result. However, when a thread sent a judging request to the sandbox box, and then send a checking request, the asynchronous communication may cause chaos in the order of the requests sequence and return results, resulting in data processing error or cause a time waste.

The message box mechanism can avoid these forgoing problems and improve the efficiency of data processing. Each information interacted with the sandbox has a unique message number. Uniformity the information which returned back from the sandbox and contained the message number, are received by the message box. Thus the system can get the correct information according to the message number. The message box mechanism realizes the unified management of messaging, and the Web side only need to prepare information and returning work, for the details of the information transmitting and receiving is transparent to the Web side.

4.5 Sandbox connecting manager

For the quantity of sandboxes interacting with the WEB side is large, design the sandbox connecting manager to manage the status of each sandbox, which would achieve load balancing between the sandboxes and improve the efficiency of the system. The sandbox connecting manager not only maintained contacting with multiple sandboxes, but also responsible for the important work as follows:

(1) Maintenance queues of sandboxes: design three queues to manage sandboxes of three different statuses-executing judgment, idle, and suspended status. In order to keep high concurrency, we must safeguard the three queues and maintain each sandbox inside the queue, to ensure the corresponding sandbox to finish its task in arranged order. When issue a Judge request, the sandbox connecting manager could preferentially and quickly get to the idle sandbox, and hand over the request to it.

(2) Maintenance the queue of judging requests: Usually the judging requests are growing rapidly, but the processing speed of each sandbox is lower than the growth rate. Therefore designs the queue of judging request to ensure that each request is not lost, and then issued the judging requests to the sandbox according to the rules of FIFO (First in First out).

4.6 Sandbox Business Manager

Each state of the sandbox (such as memory usage, judge, suspended, etc.) is constantly changed. If each client sends request to check states to their separate sandbox, especially some state information must be gotten through the network from a remote sandbox, which would be a huge and unnecessary overhead. Therefore, design sandbox business manager (JavaSandboxService) to manage sandboxes, which obtain and update the contents of the cache of the sandbox state by the uniform. The JavaSandboxService means that the increase of transparent buffer layer upon the web side, which can obtain the sandbox state through the JavaSandboxService, that isn't real-time but cached state. For the badly situation of more large sandboxes and sandbox clients, it will not cause a large amount of waste of system resources, but also a good performance experience, for the sandbox state has been cached in memory.

4.7 Performance optimization of database

Usually design of the data table has to ensure data integrity and consistency. But to OAJ system, it is so frequent to operate the database, that the performance is much higher than the integrity and consistency of data. Thus one of important design tasks is to optimize performance of the database.

(1) Not setting foreign key

The foreign key design will increase the burden of performance. Take the answer table for example: OAJ system has to insert a lot of answer records every day. If there is a foreign key constraint, every time the system has to scan to qualify the record. Especially when a table has multiple foreign keys, the number of scanning records is increase exponentially and greatly reduced the performance. With the presence of foreign keys, each operations of INSERT, DELETE or UPDATE must require more

complex mechanism to capture errors. The resources on the database server is every expensive in the environment of mass data. For OAJ system does not involve high consistency requirements, such as money, data redundancy is tolerable comparing to the cheaper storage. So according to requirements of business, each data table is not set the foreign keys, and the data consistency and integrity are guaranteed by program.

Furthermore not setting foreign key ensures the OAJ system to easily split table, which improves the query performance by using the redundancy design. Considering to the future business expansion, it will probably require deployment different service to different servers, such as deploy the backstage management, the Web side, the sandbox side to individual server. Then it is required that the servers must have separated database for themselves services. It means that the database tables storing the information are scattered in different servers, so cannot establish a foreign key constraint, but to control the data consistency and uniqueness by the application.

(2) Splitting the submit_record table

To the application system based on database centre, the most frequent operation is query, which will directly affect the system performance^[8]. For OAJ system, the records are frequently inserted into submit_record table. It is estimated that about 300 thousand monthly are inserted into this table. As time goes on, this table will become very large. And then when the user submits his codes, the system will be very slow in speed, for it need to query all contents of the submit_record table. Especially to the conditional queries without index, the retrieval speed will be unbearable for the system has to search all records of the table. To overcome this problem, OAJ system horizontal split the submit_record table. The splitting rule is depicted as follows: the record of the user submitted code are stored into the submit_recordX table, where X is the largest integer for N divided by user ID, and N is positive integer greater than 2 (usually N is 10). The N can be a estimated value according to the total number of users, to ensure the performance optimization of the database.

5 Conclusions

The OAJ system is developed using Spring + Spring MVC + Mybatis (SSM) framework, and realize functions of competing management, problem management, automatic judgment, match reports and other functions. The system takes focus on improving the judgment speed and accuracy, performance optimization to database. The simulations show that the system has the advantages of high reliability, efficiency, safety and good extensibility.

Acknowledgments

The work was supported by Science and Technology Project of Guangdong Province under contract (2014A040402010 , 2016A010101028) and Project of Enhancing School With Innovation of Guangdong Ocean University Grant by CXXL2016134. The college education practice base construction project of undergraduate teaching belongs to the ministry of education in colleges (48[2013]) Grant the department of higher education of ministry of education of China.

References

- [1] X. Du, C. Yi, Y. Wei, S. Feng and Z. Gong, Design of Automata Online Judge, *2nd International Conference on Information Engineering and Computer Science*, pp. 1-4, (2010).
- [2] Jianhua Wu, Shuangping Chen and Rongrong Yang, Development and application of online judge system, *2012 International Symposium on Information Technologies in Medicine and Education*, pp. 83-86, (2012).
- [3] S. Li, Y. Zhuo, S. Feng and C. Yi, The application of course-oriented online judge in "Data Structure", *2012 IEEE Symposium on Robotics and Applications (ISRA)*, pp. 687-690, (2012).
- [4] N. A. Tonin, F. A. Zanin and J. L. Bez, Enhancing traditional Algorithms classes using URI Online Judge, *2012 International Conference on E-Learning and E-Technologies in Education (ICEEE)*, pp. 110-113, (2012).
- [5] Wang Lei, Micro Service Architecture and Practice, *Electronic Industry Press*, pp. 75-101, (2015).

- [6] Yingsheng He, Qiang Luo, Design and implementation of kernel evaluation of Online Judge [J]. *Journal of Jishou University (NATURAL SCIENCE EDITION)*, vol.31 (6): pp.42-44, (2010).
- [7] Dingcai Li, Shaojun Qu, Zheng Hu, et al. Research on the security of online scoring system based on [J]. *technology and development of computer Windows*, (21): 204-207 (2011).
- [8] Gu zhenli, Research on Query Optimization of relational database [J]. *microcomputer information*, vol. 5X: pp. 162-164, (2006)