Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2015, 7(3):2418-2421

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Design of experimental platform of data structure based on online judge

Qin Zhang, Xinyu Li, Xiaopeng Xie and Jiangping Ou

School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

ABSTRACT

Famous Online Judge systems are provided for ACM International Collegiate Programming Contest. These systems are inappropriate for the experiments of Data Structure. The experimental platform employs LNMP as development environment. For performance, the platform can support high concurrency with least resources. For stability, the platform uses a more scalable event-driven architecture that keeps CPU and memory under low load. Besides, it is high available, supporting hot deployment, which can be started quickly and upgrade smoothly. It takes architecture technology of existing large-scale websites as references, focused on performance and user experience.

Keywords: Data Structure, Online Judge, Experimental Platform

INTRODUCTION

Data Structure is a fundamental class for computer science undergraduates, which emphasizes on practice. Data structure underlines program running time and memory dealing with mega data. Traditional programming mode, only allow students to check their own programs for syntax errors and running errors, while lacking consideration of mega data and the ability to reflect the complexity of time and space. Online judge mode was introduced to build experimental platform for data structure in order to solve those problems [1,2,3]. At present, online judge system of ACM/ICPC has the B/S and C/S two implementations. Typical PC^2 is the C/S structure of the online judge system, featuring with fast response and high security features, suitable for big game. While installation of client is needed, students can not use it anytime. These well-known online systems of B/S model including POJ of Peking University [4] and ZOJ of Zhejiang University [5], providing a good training platform for students who love programming. However, these systems are designed for competition, functionless in experimental teaching, and resulting the teachers feel inconvenient to use. It is essential to build an experimental platform of data structure.

The design of the experimental platform is based by open source code, make it easy for customized design and deployment. It takes architecture technology of existing large-scale websites as references, focused on performance and user experience. Platform provides students with the training and questions answering of programming. For teachers, it offers creating experiments or exams, submit questions and test data, download the students code and results of experiments and other functions.

1. System Architecture

Based on online judge, the experimental platform employs LNMP (Linux + Nginx + MySQL + PHP) as a development environment. For performance, Nginx is coded in C and can support high concurrency with least resources. In terms of stability, Nginx uses a more scalable event-driven architecture that keeps CPU and memory under low load. Besides, Nginx is high available, supporting hot deployment, which can be started quickly and upgrade smoothly.

System architecture can be divided into three sections: the web server, the database server and the judge server, which showed in Figure 1 as follow:

(1) Web server, as the communication window connect the outside world and system, whose interaction design determines the user experience, and its performance impacts on the system overall throughput directly. Components in web server are part into the static and the dynamic. Static contents (CSS, JS, images, etc.) using a independent subdomain, stored in a third-party CDN service, which can shorten page loading time and reduce traffic pressure on the server as well. Dynamic contents can use opcode cache such as APC, with varnish + ESI (Edge Side Includes) to achieve local cache. This method can effectively reduce the server load, while increase the response time of user inquiries. Evaluation system has lots of interaction, each user get access to different contents after to log in, the use of local caching can maximize the reuse of resources.

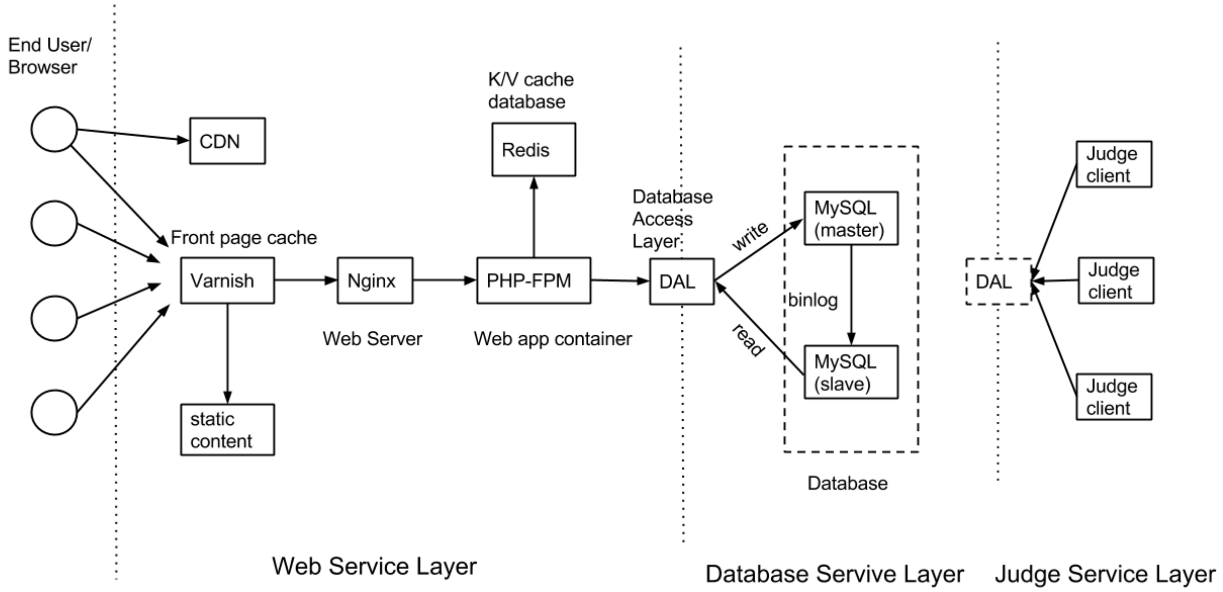


Fig. 1 System architecture

(2) Database server is the interaction channel between web server and judge server, and it is one of the factors that affect system performance. Problems, courses, contests, user code, and other core information are recorded in table of the database. Database server is implemented in MySQL, which is a small-scale relational database management system. MySQL has multiple engines for flexible options, considering that the system needs to deal with high concurrent data write and update, so it use row-level locking InnoDb engine charactering ACID. InnoDb keeps transaction isolation by MVCC mechanisms, MVCC save a snapshot for transaction when it happens. It can ensure data consistency within the transaction, but also to avoid the extensive use of locks, so as to enhance the performance and concurrency. InnoDb also has a good ability of crash recovery, to ensure the recoverability of data, providing a reliable environment.

(3) Judge server existing as a system service, do managing by upstart, the event-driven service management program, and monitoring the judgments task queue in database server at the same time. When a user submits a new answer, web server add the judge task to the task queue in the judge server, waiting for the judge server to automatically lock the task and complete. Lock can be done via static allocation of answer number or locking mechanism of database. This design ensures the scalability for the future that judge servers can be added according to the load dynamically. Judge server consists of daemon process and judging process, Figure 2 shows the flow of judge.

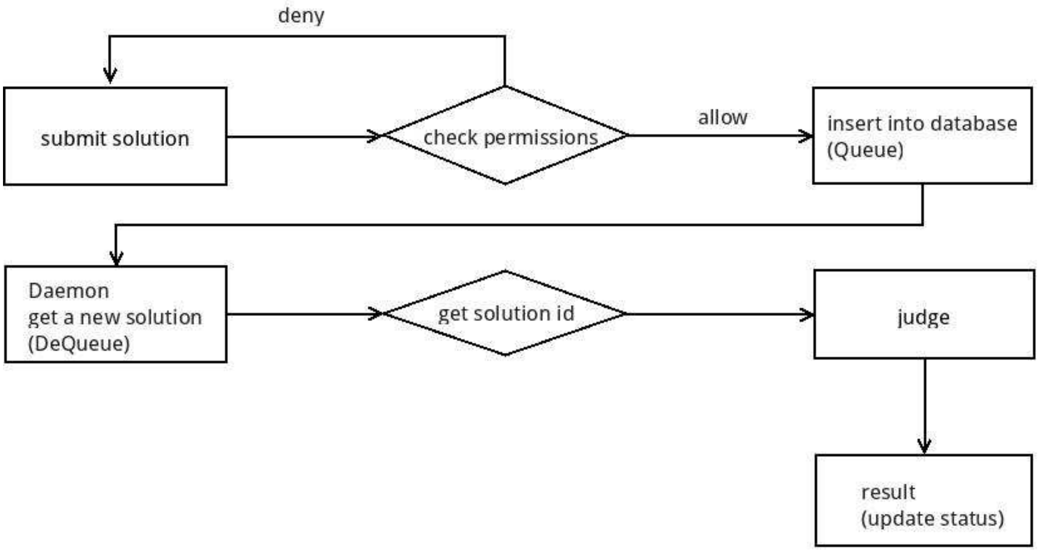


Fig. 2 Judge flow

2. Code Logical Flow

The system isolates model, view and controller, so it can make better separate presentation and logical layers, and scalability is enhanced. It can also add related modules according to needs, while make maintenance easier. In the MVC framework, errors are easier to locate. Code logical flow shown in Figure 3.

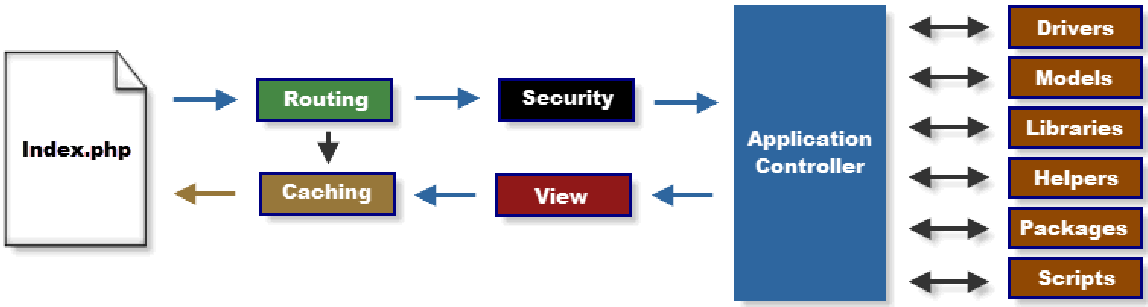


Fig.3 Code logical flow

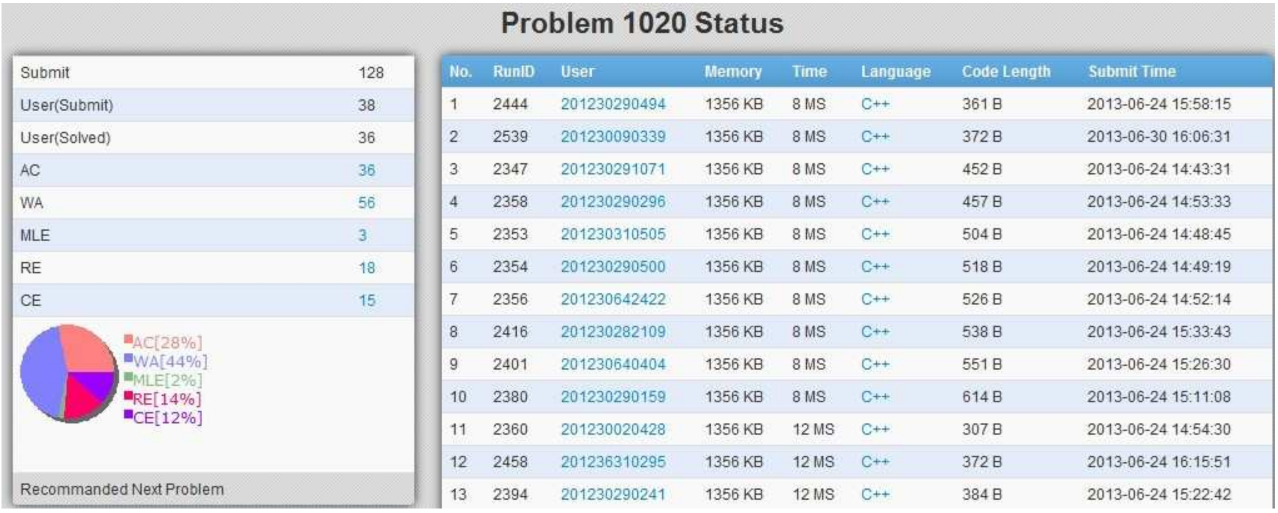
When inquiring the site root directory, web server will request index.php automatically. INDEX as the front-end controller, initializes the basic resources to run the system and the components needed to auto load in config / autoload of customized library. This alleviates the confusion that each controller repeats loading the same component, and facilitates unified maintenance and management. Subsequently, router will check HTTP / HTTPS requests to determine which controller to handle the request. Caching is used in order to speed up page loading and rendering, pages that are frequently requested and not modified frequently will automatically generate cache. If the cache file correspond to request exists, the system will pass the normal execution order, and the page will be sent directly to the browser, thus saving time and resources.

3. Functional Modules

System consists of four functional modules, namely, course, contest, user and background. Course and contest modules share problem sub-module, it consists of problem list, problem content, solution submit, status inquiries. User module implement login / logout, registration, user status display by Ajax. Background module implement management of student, problem, privilege, contest, course and news. Background module customized for teachers' needs of processing students data in batch. It can generate accounts with roster and add user group with courses in batch. Course related data could also be exported in tabular form.

The system provides a data test function based on standard programs, the teacher provide standard program, and can directly detect the legitimacy of data after adding the testing data, the error will be prompted. Problems are labeled by category, and students can search by relative knowledge points, which have guiding impact on problem solving. System maintains real-time statistical information of every student, per problem per course. There is information on the problem submitted amounts, corrects, errors, and the time of the user submit. After the due of the course unit time, the teacher can export course data analysis table by one click for further processing.

Figure 4 shows the results of some problem



Problem 1020 Status

Submit	128
User(Submit)	38
User(Solved)	36
AC	36
WA	56
MLE	3
RE	18
CE	15

Recommanded Next Problem

No.	RunID	User	Memory	Time	Language	Code Length	Submit Time
1	2444	201230290494	1356 KB	8 MS	C++	361 B	2013-06-24 15:58:15
2	2539	201230090339	1356 KB	8 MS	C++	372 B	2013-06-30 16:06:31
3	2347	201230291071	1356 KB	8 MS	C++	452 B	2013-06-24 14:43:31
4	2358	201230290296	1356 KB	8 MS	C++	457 B	2013-06-24 14:53:33
5	2353	201230310505	1356 KB	8 MS	C++	504 B	2013-06-24 14:48:45
6	2354	201230290500	1356 KB	8 MS	C++	518 B	2013-06-24 14:49:19
7	2356	201230642422	1356 KB	8 MS	C++	526 B	2013-06-24 14:52:14
8	2416	201230282109	1356 KB	8 MS	C++	538 B	2013-06-24 15:33:43
9	2401	201230640404	1356 KB	8 MS	C++	551 B	2013-06-24 15:26:30
10	2380	201230290159	1356 KB	8 MS	C++	614 B	2013-06-24 15:11:08
11	2360	201230020428	1356 KB	12 MS	C++	307 B	2013-06-24 14:54:30
12	2458	201236310295	1356 KB	12 MS	C++	372 B	2013-06-24 16:15:51
13	2394	201230290241	1356 KB	12 MS	C++	384 B	2013-06-24 15:22:42

Fig.4 Problem status

CONCLUSION

The experimental platform of Data Structure based on online judge improves students' programming skills. At the same time, it makes teachers convenient to know the abilities of students. Next, the experimental platform needed to get feedbacks, keeping find out and solve problems, enrich the functions. Then we will enlarge problem repository and classify the problems more clearly.

REFERENCES

[1] Jian-min Han, et al.: *Computer Education*, **2013,** No.10, p.103-107.
[2] Ying-jie Wu et al. Computer Education, **2010,** No.4,p.53-55.
[3] Qin Zhang et al. *Journal of south China university of technology (social science edition) supplement*,(**2012**)No14, p.75-77.
[4] Peking University Online Judge System http://poj.org/
[5] Zhejing University Online Judge System http://acm.zju.edu.cn/onlinejudge/