Analysis of online compiler

Sankesh Pathak
sankeshpathak@yahoo.com
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

Saif Husain
saifhusain990@gmail.com
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

Ankush Nanglia
anangliya@gmail.com
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

Nikhil Lund
nikhilloond@gmail.com
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

Ketankumar Fulbandhe
ketanfulbandhe@gmail.com
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

Meghna Deshmukh
madeshmukh@mitra.ac.in
Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra

ABSTRACT

With the development of online education, the pattern of education which based on the Internet obtained the general recognition of the society. People prefer to learn and share the open courses on the Internet in universities or colleges all over the world. However, the courses opened online can support the massive peoples to learn online, but couldn't support them to program on the network, and judge the program on the network. In order to support more and more students to complete the experiments of the online courses, automatic judging system about software experiments and hardware experiments are needed to develop to ensure the correctness of the design of experiments. The Background Evaluation System is in an online environment, it can connect different experiment systems to finish different courses. It also can simulate the hardware experiments online and test hardware experiments automatically. Online judge system, can solve much of the problems.

Keywords: Programming, Methodology, Configuration, Judge, Compiler, Grading, Queue, Mechanism.

1. INTRODUCTION

Programming is one of the major skills taught in the Department of Computer Science. Virtually every Computer Science module involves some form of programming[1]. Needless to say, some Computer Science modules require more programming than others. Programming is a skill acquired through practice. To help students in this respect, programming assignments are given to students. The number of programming assignments reveals how much emphasis on programming a module has[2]. Most of the higher-level modules would assume that students are already comfortable with programming in virtually any programming language, and thus has only one major programming project that emphasizes on the topic being covered, such as client-server networking and state space searching[3]. However, modules that emphasize programmings, such as Programming Methodology, Data Structures and Algorithms, and Competitive Programming, aim to nurture students in programming and have many programming assignments with questions of varying difficulties. At the time of learning languages like C, C++, Java … Etc[4]. The beginners had a number of problem like to install compilers at desktop-pc. Some time student had to face the problem of configuring the compiler. So that they can’t get execute the program successfully. And another problem is that if the user wants to compile the program on any other system they require again all processing to install configuration problem[5]. Security is one more problem of the user or programmer. Because due to the viruses or some time not properly handling the application the source file may be damaged or lost by the system crash.

2. RELATED WORK

A. Online Judge

An online judge is an online system to test programs in programming contests. They are also used to practice for such contests. Many of these systems organize their own contests. The system can compile and execute code, and test them with pre-constructed
data[6]. The submitted code may be run with restrictions, including time limit, memory limit, security restriction and so on. The output of the code will be captured by the system and compared with the standard output. The system will then return the result. When mistakes were found in a standard output, re-judgment using the same method must be made. Online Judges have rank lists showing users with the biggest number of accepted solutions and shortest execution time for a particular problem.

B. Server Based Compiler

Server Based Compiler is a compiler located on Server. A thin client or a web browser allows the compiler to be controlled from a location elsewhere around the world.

C. Working of online judge

This is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort[7]. The paper aims to describe an online compiler which helps to reduce the problems of portability and storage space. The ability to use different compilers allows a programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover, a web-based application can be used remotely throughout any network connection and it is platform independent. The errors/outputs of the code are stored in a more convenient way. Also, the trouble of installing the compiler on each computer is avoided. Thus, these advantages make this application ideal for conducting examinations online.

D. Automated Grading

To attempt to alleviate the problems outlined in the previous section, we intend to have an online judge system mounted on a dedicated machine. It knows about all the programming assignment questions. Students submit their programs electronically to this machine. The system will check the correctness of the submitted programs. A dedicated machine would be reasonably secure since it would have its permissions set correctly. With it, we prevent some problems that might occur when an unsuspecting grader executes a student’s malicious program. This dedicated machine would ideally have all the permission settings set correctly, such that it is impossible for a submitted program to do the malicious actions as described previously. This machine would also handle the killing of the slow programs. The model we are proposing is as follows. Students submit their programs. The machine receives, logs, and grades them. With all the permission settings set correctly, any attempt to do malicious actions such as removing all files would not succeed. Note that even if it succeeded, the files owned by staff members would not be affected since this would be a dedicated machine. As we have noted earlier, the programs we are interested in grading are algorithmic in nature, and can thus be used as if they were filters. They take in some input, do some algorithmic processes, and write out the results[5]. Our system will provide the input through the standard input handle and take in the output from the standard output handle. The programs themselves will not be looked at. Instead, their output will. A program will be accepted only if it produces the correct output within the specified time limit. It will be rejected otherwise. A program’s output will not be examined at all if the program does not terminate within the time limit. A question will be phrased in terms of how the input looks, and what each of the components means; how the program processes the input, and how the output should look. Parsing of input is usually not the main focus of the problem, the input format is usually very simple and the input data is guaranteed to match the specifications. In other words, there is no need to check, for example, whether a number in the input is indeed within the prescribed range. In order to help students understand the problem and to test their solution, a sample test set, consisting of the input and the corresponding output, is provided. This would instantly give an idea of the format of the input and output. In order to grade their solutions, however, there have to be other pairs of secret test sets. The secret test sets would weed out programs that would always display the sample output. The secret test sets have to be secret. Otherwise, students would submit a program that would check whether the input is that of the first test set. If so, the output of the first test set would be displayed as a result. The same for the rest of the test sets would be done. There are two claims of using multiple test sets. One is that if a program fails on any one test set, it is not a working program. The other is that each test set is there for a specific purpose —perhaps to test different boundary conditions. Imperfect programs that passed some of the test sets are to be given partial credits. We go by the second claim, with the idea that when the first is desired, we can always use just one test set with multiple test cases, and amend the input/output specifications in the question that all of the test cases must be processed.

3. METHODOLOGY

A. Functions and Flow

This interface is simple and intuitive. An interface defines the communication boundary between two entities, in this case between the students and the application. The concept is a completely online application. The virtual editor present on the web site, the user first registers the site and then gets the ID and PASSWORD.

After the student identification, s/he can choose a section from options as given below
The Code editor is the text area where the user can write or paste his/her source code. In code editor, s/he must choose the programming language with which s/he is going to work. This can be performed from the selection language menu. Once the programming language is chosen, the screen of code editor will be ready for that particular language code. The tabs menu locates at the top of the screen, offers users the navigation through the home page sections like an editor, the files, the tutorials and other sections. The users need to either write code on code editor provided on the platform or upload their solution in the form of a file. After submitting the source code file the user can see the verdict in submission queue area. Ranks will be given on the basis of contests. The problems solved in practice sessions or in academic assignments unless assignments are in form of the contest will not be considered for ranking. The rank list will be of IOI Style as described below, you will be ranked on the basis of the score that you get. There are no penalties for wrong submissions, so you can submit solutions as many times as you'd like. However, only your best correct submission will be considered. Ties are broken according to the times each participant reached that score.

B. Submission Queue

After submitting source code from code editor or directly uploading the solution from a file, the source code is put in a queue for judgment. Submission queue is temporary holding location where we can see the verdict to the code that user has submitted. For a user to see his/her code verdicts only he can opt for my submission page where only his submissions are present.

C. Judgment Mechanism

A program may be judged as Compile Error if it does not compile properly. The user is notified of such cases. For Java, we had an extra error message, Class named main not found since we have to fix the class name because there is no easy way to determine what the class name is. If a program compiles, the compiled program is executed once for each test set to generate each of the following individual judgments:

- Accepted: the program terminated within the time limit and produced correct output in the correct format.
- Presentation Error: the program terminated within the time limit, and produced correct output, but in the wrong format, for example, with an extra blank space.
- Wrong Answer: the program terminated within the time limit, but produced incorrect output.
- Runtime Error: the program is crashed during execution.
- Time Limit Exceeded: the program did not terminate within the time limit, and it was killed with signal 9 (SIGKILL).

D. Security issues

I have noted previously that we can receive a malicious program at any time. We have also stated that scanning source files for unwanted tokens such as unlink may not work because:

- It is far too easy to forget about a keyword. A hacker knowing the loophole might just make use of it.
- It is very difficult to introduce a new language to the server. More often than not, the language has to be thoroughly learned in order to extract a list of keywords that are unsafe. Again, it is far too easy to forget about a keyword.
- It is virtually impossible to introduce an interpretive language that has the ability to execute a string it creates during runtime. In Perl, one would construct a string that has malicious commands and then evaluate this string. Keyword banning will not identify it because the string can be constructed in a very obscure manner, perhaps by first constructing the commands that construct this string into another string.

We adopted a different approach. We wanted to be able to introduce new programming languages. Sometimes it may be difficult, but it should be possible. The above approach would not work on languages like Perl.

We made our environment very secure, and only execute the programs under that secure environment. We do not care about what the program does. But when it attempts to do such unexpected actions, it should fail.

Ideally, this involves a major rewriting of the kernel, to weed out the unwanted functions (such as networking) in order to prevent a program from doing something nasty with these functions.
E. Time limit

We limit the execution time using limit as above. The program is allowed to execute until its user time and system time total more than one more second than what we set. This is fair because it does not count the time when other processes, not necessarily related to judging, are running. One such process is the web daemon [5]. When the program overtimes, it will be terminated using the SIGKILL signal, which guarantees the termination of the process. However, this has its problems. Consider a user who submitted a program that does sleep. The program would have slept for a day without consuming any user time, and consequently, the judge would also have spent a day waiting for the program to terminate.

F. System load

Since the judge is running on a dedicated machine, it is very idle. The judge’s existence will hardly overload the system [7]. We set the process priorities such that the judging system does not interfere with other processes running on the server. We found that the system works well by setting the judge’s real-time-counter program’s priority very low, and the program is judged to be slightly higher [8].

4. ACKNOWLEDGMENT

I would like to express my gratitude to all who gave me the opportunity to prepare this seminar. I would like to thank my project guide, Prof. M. A. DESHMUKH for her kind suggestion, inspiration, and guidance during preparation of this seminar. I express my deep sense of gratitude towards Prof. Dr. V. M. Deshmukh, Head of the Computer Science Engineering Department for granting me permission to commence the seminar. I would also thank the faculties of the Department of Computer Science & Engineering, for their kind cooperation and encouragement which helped me in the completion of this seminar. I owe an incalculable debt to all staffs of the Department of Computer Science & Engineering for their direct and indirect help. At last, I express my gratitude to my colleagues for their support and help.

5. REFERENCES