CRJudge: A Webapp to Judge Source Codes

Ajay Kumar Saini[1], Aviral Modi[2], Vinita Chauhan[3]

[1,2] Student, Dept. Computer Science & Engineering, IMS Engineering College, Ghaziabad, India
[3]Asst. Professor, Dept. Computer Science & Engineering, IMS Engineering College, Ghaziabad, India

Abstract - *This report describes an implementation of a web app for automatic program assessment and grading, named as CRJudge. The web app is highly useful for instructors as well as students, which enriches the learning experience and makes program assessment and grading faster, consistent and unbiased. A variety of learning material can be made accessible to students by the app. It is equipped with various grading strategies and plagiarism detection addons. The app is tested against various internet threat and capable of handling malicious programs gracefully.*

Key Words: online judge, assessment tool, programming contest, automatic grading, programming practice

1. INTRODUCTION

The software industry is growing large and highly competitive. In order to compete and survive, the industry requires professionals who have great programming skills to develop a state-of-the-art technology. To hone such skills, students must be trained and monitored continuously by professors or instructors as a part of academic curriculum. This report presents the description of such a web-based tool which can be helpful to solve the purpose. The tool enriches the learning experience and makes programming practice much easier and interesting. The tool allows instructors to create new problems and contest, assess and monitor the performance of students. It offers a wide range of programming language support to students which allow them to write code in the language of their choice and a variety of learning material is made available to students to learn and practice. It will encourage students to practice more and more since judgment is unbiased and swift. However, it may become frustrating for a student if he or she fails to solve the problem repeatedly.

2. THE PROBLEM

The conventional way of judging programs in programming classes is manual checking by the instructor.

Manual checking of programs has certain flaws [1] which are as follows:

1. Difficult to calculate and compare running time of a program.
2. No guard against malicious code.
3. The instructor may judge solutions inconsistently depending mood.
4. Correctness of judgement is not certain.
5. The manual judgment of a solution is slow.
6. Rejudgement of the solution is difficult.
7. More manpower required.
8. Manual judgment process is slow.
9. Less participation and less contests or assignments.
10. Tough to analyse the performance of a student.

3. THE CRJudge

CRJudge is a web app to automate the process of assessing solutions for programming problems and reporting the performance of the student. It works only on Linux platform which can be accessed by any modern browser. It employs the use of numerous shell scripts and RESTful API. The tool will encourage the self-taught programmers to learn and sharpen their programming skills. The instructors are facilitated with automatic testing, judging, and plagiarism detection, which saves their precious time and allow them to train more and more students.

The application can perform following operations:

- The instructor can create ACM-ICPC style problem statement and submit corresponding test cases and tag them.
- The instructor can create ACM-ICPC style contest and send a notification to the target audience.
- Students can view problems and upcoming contests and filter them by difficulty level and required knowledge level.
- Students can submit a solution to the problem in the language of their choice and also view solution of other students.
- Students can participate in contests and get supporting material after the contest is over.
- Students can keep track of their ranking and performance.
- It is also capable of detecting plagiarism in solutions using the tool provided by Stanford university, name MOSS [2] (Measure Of Software Similarity).

4. IMPLEMENTATION

4.1 Platform

The application runs on a virtual Linux machine provided by amazon cloud service. On top of Linux, apache HTTP server, and MySQL database is used. One or more similar machine can be dedicated for judge, this enables graceful handling of system crashes caused due to malicious codes.

4.2 Programming Language

C++, PHP and BASH scripting language.

4.3 Database Model

Figure 1 depicts the database model used in application. It has following entities:

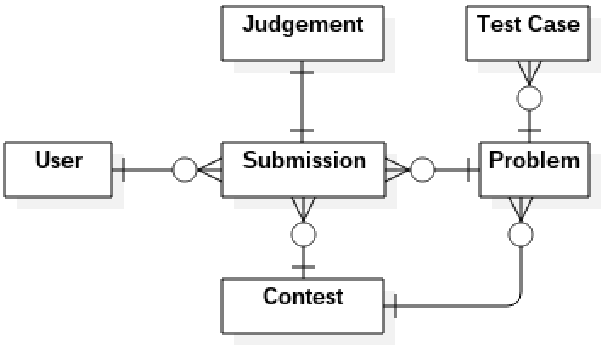

Fig - 1: Database Model

User- This represents 'user' entity. User privileges are defined by the roles assigned to it. Only following roles are allowed in application:

1. *Admin* has all privileges.
2. *Instructor* can create and modify contests, problems, judge and re-judge solutions, and send notifications to students.
3. *Student* can view all contest and problems, submit solutions and view his/her progress.
4.

Contest- A contest may also be referred as assignment. A contest can be created by an instructor or admin.

Problem- This entity represents a problem in ACM-ICPC style. It has problem title, description, input format, output format, and sample test case as attributes.

Submission- This entity represents the solution submitted by a student for a problem containing source file and metadata as attribute.

Testcase- This entity represents test cases which is used by judge to test the submission. A problem may have one or more test case. Each test case has a pair of file, one for inputs and other for expected outputs. A test case may also have custom configuration file used by judge.

Judgement- Each submission will have exactly one judgement describing the outcome of the submission. Judgement as attribute called as verdict which can possess following values [2]:

- MLE (Memory Limit Exceeded)
- TLE (Time Limit Exceeded)
- WA (Wrong Answer)
- CA (Correct Answer)
- RE (Runtime Error)
- CE (Compilation Error)
- DJ (Denial of Judgement)

4.4 Queues

System has one queue called as submission queue. A soon as soon as a solution is submitted by a student, it enters the submission queue. From the queue submission is dispatched to the judge.

4.5 Judge Pipeline

After a submission is dispatched to judge it goes through a three staged pipeline (Figure 2).

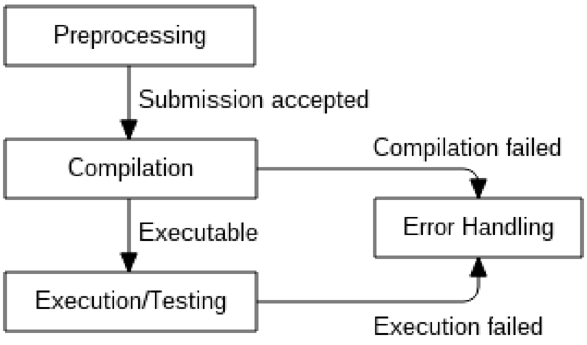

Fig - 2: Judge Pipeline

Preprocessing- At this stage the submission is rejected if it is duplicate. If the judgement is rejected it results in verdict DJ (Denial of Judgement). After that certain code stubs may be added if required.

Compilation- At this stage the code is compiled using appropriate compiler installed locally. Compilation may result in verdict CE (Compilation Error) Interpreted language, such as Python, do not go through this stage.

Execution- At this stage the executable file is run and tested against test cases. The executable is run securely in a simple yet powerful sandbox. This stage may result in CA (Correct Answer), WA (Wrong Answer) RTE (Runtime

Error), TLE (Time Limit Exceeded), MLE (Memory Limit Exceeded).

4.6 Sandbox

A user may submit malicious code intentionally or unintentionally. To handle this situation, the executable in run as unprivileged user under a sandbox. The sandbox blocks all system calls made by executable file except exit(), sigreturn(), read(), and write() to already-open file descriptors. If it attempts any other system calls, the sandbox will terminate the process with SIGKILL. This is made possible using SECCOMP mode [3] (short for secure computing mode) provided in Linux kernel. This mode is used in major commercial sandboxes, but they are not suitable in our case because of their overheads. This sandbox is made only for C/C++ codes or native executables.
To sandbox Java executable Java Security Manager is used to deny access to system resources such as files, networking etc.

4.7 Resource Throttling

There are mainly two resources that needs to be throttled - CPU time and memory, other resources are denied completely. This is done using CGroups [4] (short for control groups) provided in Linux kernel. CGroup is a kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.
Another simple solution is to use a Linux utility called *ulimit.* ulimit uses a configuration file and is easy to set up as compared to CGroups but CGroups is found to be more dynamic.

5. APPLICABILITY

CRJudge is highly useful in scenarios where we need to train or test logical programming skills of a person. In academic institutions where programming is taught to students CRJudge can be used as automatic grading system.

Competitive programming is another use case where CRJudge can be employed as a portal to accept and judge solutions and rank participants by their performance.

Companies hiring software professionals can use CRJudge to automate selection procedure. In this case the company can save their precious manpower and money.

6. CONCLUSIONS

We have successfully implemented and shown the working of CRJudge. CRJudge overcomes all flaws in manual judging or grading system and offers many advantages such immediate response, unbiased verdict, organised supporting material etc. Most importantly it helps saving precious time of instructors and students. All these advantages makes learning process much easier and tempting for students who wants to master their programming skills. However, students of other field may not be benefited from this app.

REFERENCES

[1] AndyKurnia, Andrew lim, and Brenda Cheang, “Online Judge”. https://www.researchgate.net/publication/222285496_Online_Judge1.

[2] “MOSS (Measure of Software Similarity)”. https://theory.stanford.edu/~aiken/moss/.

[3] “Seccomp”. https://en.wikipedia.org/wiki/Seccomp.

[4] “CGroups”. https://en.wikipedia.org/wiki/Cgroups.