

Secured Programming Contest System with Online and Real-time Judgment Capability

Ahmed Shamsul Arefin, Muhammad Arifur Rahman, Shusmita Anwar Sharna, Samiran Mahmud[†],
Dr. M. Kaykobad[‡]

Dept. of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh.

[†] Dept. of Computer Science and Engineering, DUET, Dhaka, Bangladesh.

[‡] Dept. of Computer Science and Engineering, North South University, Dhaka, Bangladesh.
asarefin@acmsolver.org, arif_ju@yahoo.com, sasharna@bdonline.com, mahmud_samir@yahoo.com,
kaykobad@northsouth.edu

Abstract

This work presents an online judge system to be used in programming contests. Such a system must provide a secure execution environment that can execute arbitrary code submitted by any programmer, without compromising confidential information or system stability. This system has been implemented in C#.NET language and can judge programs written in this same language, but has been projected with extensibility for new programming languages in mind. It has the capability of arranging simultaneously a contest in a real-time environment (Local/ICPC like contests) and online programming contest.

Keywords: Programming Contest, ICPC, Secure Execution Environment, .NET, C#, Online Judge, UML, SQL.

I. INTRODUCTION

The programming contest is one of the most dependable events for the evaluation of performance of a computer programmer. For this performance evaluation we need a judge system that will automatically judge the users' submitted programs and create a ranking among them based on the correctness and submission time of the submitted programs. This system must be able to receive a program code, compile it and judge it for its correctness and then update the scoreboard of the contest. Our proposed system has some improvement over the currently available judging software like PC2 or NETJUDGE (used by University of Valladolid Online Judge) as it creates a "sandbox" environment: which will isolate the contestant's process from the rest of the system to prevent cheating and malicious code. We have used an improved approach for time management and memory measurement. Our system is developed in C#.NET.

II. CURRENTLY AVAILABLE CONTEST SYSTEMS

There are some programming contest judge software that are available over the internet. Some of them are:

PC²: PC² was developed at California State University, Sacramento.

This system automatically timestamps and archives submitted runs, maintains and displays current contest standings in a variety of ways, and allows the judges to retrieve and re-execute archived runs. It also provides a mechanism for contestants to submit clarification requests and queries to the judges, and for the judges to reply to queries and to issue broadcast bulletins to teams. In addition, PC² supports contests being held simultaneously at multiple sites by automatically transmitting contest standing information between sites and generating a single contest-wide standings scoreboard at each remote site.

Configurable options allow the contest administrator to tailor the system to specific contest operations. For example, the number of teams, problems, and languages in the contest; the scoring method being applied; which problems are handled by which judges; whether teams are automatically notified of the result of a submission; and the frequency of automatic scoreboard updates are all configurable. There are also mechanisms provided for editing the internal scoring database, and for recovering from various types of soft and hard errors. The system is designed to allow teams to use any language development tool that can be invoked from a command line and generates an executable file. The most recent version, V8, is written in Java (using IBM VisualAge for Java) and is intended to run on any Java-enabled platform, including Windows 95/98/ME/NT/2000/XP and a variety of Unix systems including Solaris, Linux, and FreeBSD [9]. PC² Software is only capable of arranging real-time local/ regional programming contests, but it cannot host online programming contests.

NETJudge : An online judge system developed and used by University de Valladolid at Spain. This is a robot judge and can generate automated rank-list. This software is hosted with Apache Web-server and programmed with PHP, PERL and My-SQL [4]. But this judge software is used to host online programming contests and seldom used for local contests at local intranet.

And a few other universities like Ural, Zhejiang universities worldwide uses different frameworks and libraries to judge their contestants and programmers.

III. FRAMEWORK FOR JUDGE SYSTEM

In our proposed system, we have suggested some improvement over the currently available judge software. We have focused on the following improvement:

- Implementation of the Judge System with C# .NET Architecture.
- Creation of a safe environment, where source codes can be executed safely and cannot do any harm to system stability.
- Use of C# Programming Language in Contest Systems. Also it has been projected with extensibility for new programming languages
- Creating a Web-Interface for the system which can simultaneously host a contest in a real-time environment (Local/Real-time contests) and online programming contests.

IV. THE .NET FRAMEWORK

.NET is a very vast framework consisting of many pieces. It is based on following principal parts.[5][7]

Common Language Runtime (CLR) : CLR actually is a very intricate piece of software. It is not IDE (Integrated Development Environment) but rather it is a very intelligently written software to enable programmers to easily develop their software without getting involved much in the technical details. CLR is a very well managed and safe software development environment. It is the execution engine for .NET Framework applications. It provides a number of services, including

- Code management (loading and execution)
- Application memory isolation
- Verification of type safety
- Conversion of IL to native code
- Access to metadata (enhanced type information)
- Managing memory for managed objects
- Enforcement of code access security
- Exception handling, including cross-language exceptions
- Interoperation between managed code, COM objects, and pre-existing DLLs (unmanaged code and data)
- Automation of object layout
- Support for developer services (profiling, debugging)

Microsoft Intermediate Language (MSIL): is the CPU-independent instruction set into which .NET Framework programs are compiled. It contains instructions for loading, storing, initializing, and calling methods on

objects. Combined with metadata and the common type system, MSIL allows for true cross-language integration. Prior to execution, MSIL is converted to machine code. It is not interpreted.

Remote Procedure Call (RPC): RPCs have been implemented in a broad variety of ways. In the .NET framework now includes a new generic method for RPCs called Web Services. Web Services allow an application to call a function in another application running anywhere on the Internet. It uses SOAP (Simple Object Access Protocol) to communicate over the net.

C# : C# is the brand new language which is as easy as Visual Basic and is as powerful as C++. It has deepest roots in C++ with which it shares its system. It is completely object oriented.

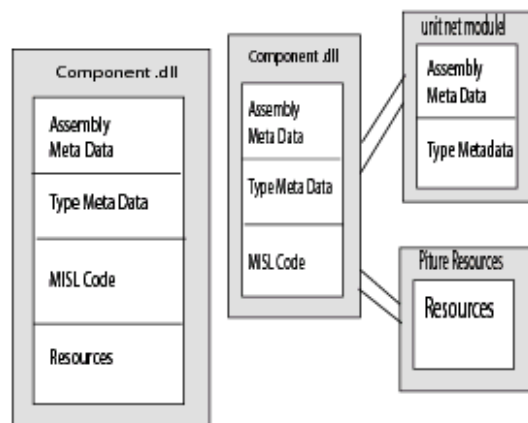


Fig. 1 .NET Framework

V. USE CASE DIAGRAM OF THE JUDGE SYSTEM

Figure 2 shows a Use Case representation of our system.

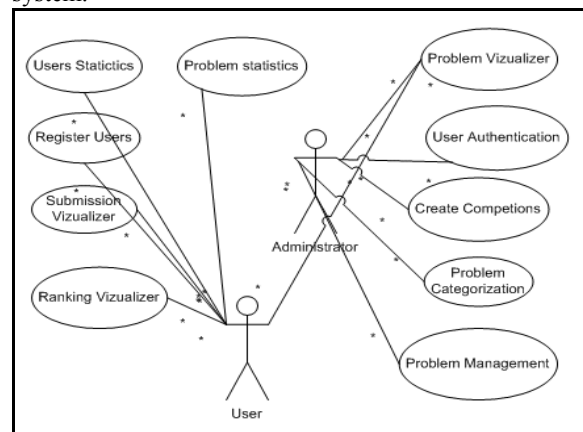


Fig 2. Use Case diagram of Online Judge

VI. SYSTEM SPECIFICATION

Now we are going to describe our overall system design through Unified Modeling Language (UML) [8]. The Unified Modeling Language™ - UML - is Object Management Group's most-used specification, and the way the world models not only application structure,

behavior, architectures and data structure. We have divided the components of our system into several parts, such as, problem set and input-output files and problem category, contests system, language set and compiler set, submission system, user registration, statistics and information.

A. Class Diagram Of Judge System

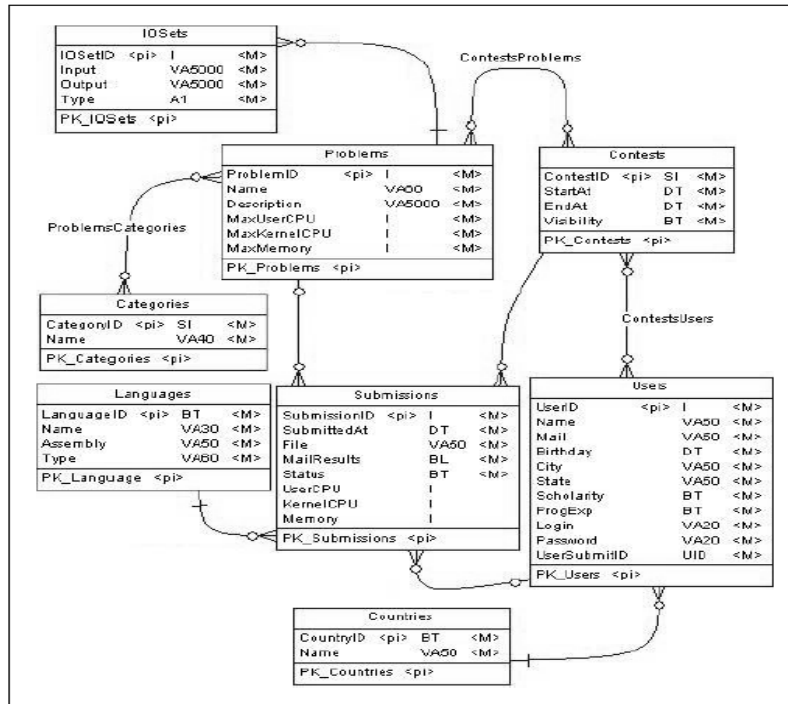


Fig 3. Component model with Class Diagram.

B. Web Interface Model Of Judge System

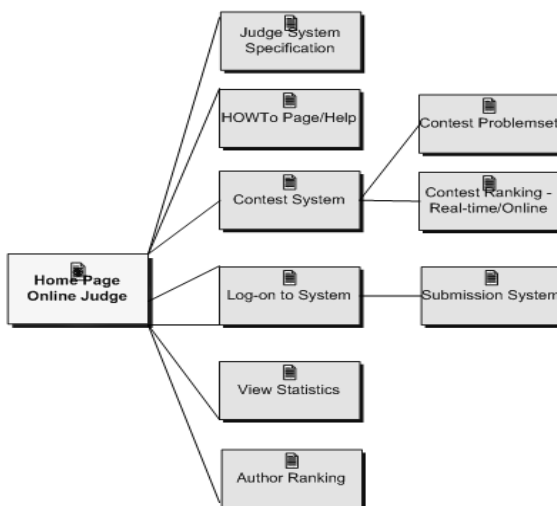


Fig 4. Conceptual web model for Judge System

C. Implementation

We have used MSDE 2.0 and Microsoft SQL Server 2000, Enterprise Edition [6]. Modular programming approach improved our system performance, for example in finding contest list by user id :

```

CREATE PROCEDURE ContestsListByUser
@User ID INT
AS
SET NOCOUNT ON

SELECT c.ContestID, c.startAt, c.EndAt, c.Visibility
FROM Contests c INNER JOIN ContestUsers cu ON c.ContestID
= cu.ContestID
WHERE cu.UserID = @UserID ORDER BY start at DESC
    
```

For starting the Judge Server :

```

private Judge judgeServer;
public OnlineJudge() {
judgeServer = new Judge();
}
    
```

```
protected override void OnStart(string[] args) {
judge Server.Start(); }
protected override void OnStop() {
JudgeServer.Stop(); }
protected override void OnCustomCommand(int Command) {
if (command == 128)
judgeServer.JudgeRequest(); }
```

For Submission compile:

```
public class Submission {
.
.
.
private void SignalJudge() {
ServiceController judgeService* = new
ServiceController("OnlineJudgeService");
try {
judgeService.ExecuteCommand(128);
}
catch {}
}
}
```

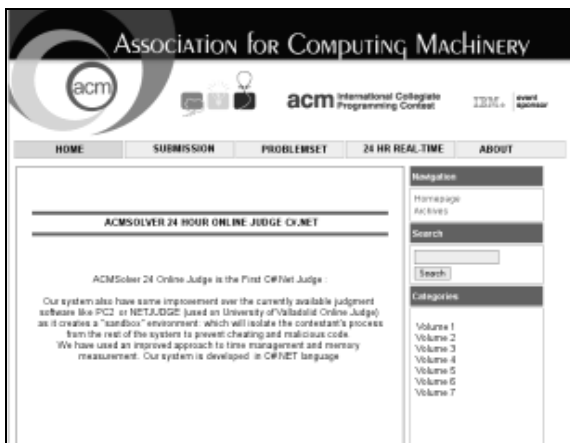
Identification will be done by a unique userID.

```
[assembly: OnlineJudgeIdentification(
UserID="B63C5208-813A-468F-9E6F-7C58ABBB10A7",
ProblemID=2,
ContestID=5)]
```

VII. WEB INTERFACE RESULTS

We have designed a distinctive user interface for our judge system. It has two parts: System Admin (for problem set administration and input-output specifications) and submission process for the users.

Homepage for proposed Online Judge:



User Registration Form

Name	Arefin
E-mail	asarefin@acmsolver.org
Date of Registration	01/11/05
City	Dhaka
State	
Country	Banglade
Education	<input type="radio"/> High School/College <input checked="" type="radio"/> Graduate <input type="radio"/> Masters <input type="radio"/> Doctor
Experience (Yrs)	4
Login Name	acmsolver
Password	*****
UserID	123477LP
Submit	

Author Ranklist :

SubmissionID	Autor	Problema	Submetido em.	Ling	Status	CPU usuario (ms)	CPU priv. (ms)	Memória (bytes)
67	Marcos Kirchner	Seqüência de Fibonacci (1)	26/11/2004 02:11:04	C#	AC	35	0	343821
10	Marcos Kirchner	Seqüência de Fibonacci (1)	26/11/2004 01:41:05	C#	WA	470	120	10096640
9	Marcos Kirchner	Seqüência de Fibonacci (1)	26/11/2004 01:39:09	C#	AC	450	140	10199040
8	Marcos Kirchner	Seqüência de Fibonacci (1)	26/11/2004 01:37:22	C#	BF	250	80	9359360
7			26/11/2004 01:36:57	C#	SE	0	0	0
6	Marcos Kirchner	Seqüência de Fibonacci (1)	26/11/2004 00:37:29	C#	WA	460	130	10108928
5	Marcos Kirchner		26/11/2004 00:24:36	C#	SE	0	0	0
4	Marcos Kirchner		26/11/2004 00:09:35	C#	SE	0	0	0

Problem Set Administration:

Name:	<input type="text"/>
Description	<input type="text"/>
MAX CPU TIME	<input type="text" value="5000"/>
MAX CPU TIME (LB)	<input type="text" value="200"/>
MEMORY USAGE	<input type="text" value="104857600"/>
SUBMIT	<input type="button" value="Add Problem"/>
INPUT	<input type="text" value="5"/> <input type="text" value="10"/> <input type="text" value="0"/>
OUTPUT	<input type="text" value="6"/> <input type="text" value="89"/>

VIII. PERFORMANCE ANALYSIS

We have been able to host our newly designed system on a Windows 2000 Server with a Real IP over a TCP/IP protocol. We submitted same source code on Several Judge software and measured times in milliseconds.

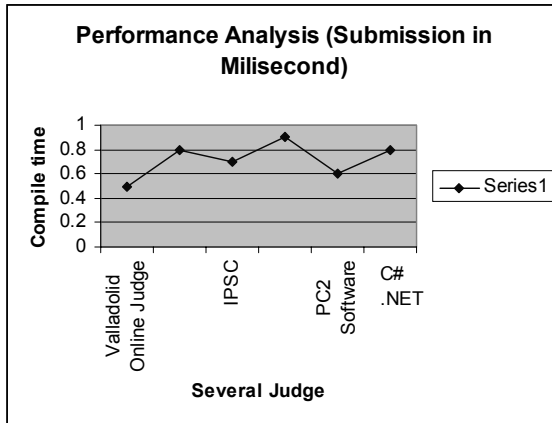


Figure 4. Performance Analysis of C# .NET Judge

IX. CONCLUSION

The progress so far we have obtained in our online judge software using .NET framework is noteworthy. Our system can be used with ACM International Collegiate Programming Contests and all other Online Contests such as IPSC (Internet Problem Solving Contest) [2] and Real Time Contests like NCPC (National Computer Programming Contest). At this moment our system works with C#, C, C++, Java and Pascal Programming language. As we have designed compiling system as a different module so, we have options for future extensions for further addition of languages.

X. ACKNOWLEDGEMENT

In preparation to this paper, we are grateful Chittagong University of Engineering and Technology (CUET) web authority for helping us to use their Dual Processor Server System and a Real IP to connect our Online Judge site to the Internet. Further thanks should go to Mr. Shahriar Manzoor, Judge, ACM/ICPC World Finals, 2003-2005 for reviewing this paper and Valladolid Online Judge team for the online judge they are maintaining at University of Valladolid, Spain from 1997 and Prof. Miguel A Revilla.

REFERENCES

- [1] ACM International Collegiate Programming Contest Homepage <http://www.acmicpc.org>
- [2] Internet Problem Solving Contest (IPSC) web page: <http://ipsc.ksp.sk/>

- [3] S. Manzoor "Common Mistakes in Online and Real-time Contests" , ACM Crossroads Student Magazine, Mid-Summer 2001, 7.5
- [4] C. Marcelino, R. Miguel, "The Valladolid Online Judge Host and Software", <http://acm.uva.es/p>
- [5] MICROSOFT CORPORATION. Microsoft .NET Framework SDK Documentation.
- [6] MICROSOFT CORPORATION. SQL Server Books Online, 2004.
- [7] Microsoft Developers Network (MSDN) Library, ".Net Framework", <http://msdn.microsoft.com/>
- [8] OBJECT MANAGEMENT GROUP. UML <http://www.uml.org>
- [9] M. Richmond, D. Bosley, P. Meyers, "An Overview of PC2" California State University, Sacramento (CSUS), <http://www.ecs.csus.edu/pc2>
- [10] Steven S. Skiena, Miguel A. Revilla, "Programming Challenges: The Programming Contest Training Manual". New York: Springer, 2003.