Cloud Based Code Studio

Prof.Mrs. Prajwal Gaikwad[1], Sujay Hamane [2], Sarang Kulkarni[3], Rahul Deore[4], Bhagwan Chavan[5]
[1, 2,3,4,5] Department of Computer Engineering,
AISSMS Institute of Information Technology,
Pune University, India.

Abstract – As today's world is said to be of Internet, all the things are fast and online. Currently single based IDE are used for particular languages. Also user was unable to access same document at same time. So we creating a cloud where can store all our documents safely and can access those documents from any location as well as writing in same document at same time from different places. Here user can also provide access rights to his /her own document to who can modify or delete or access the document located in the cloud. Along with this to implement a concept of multiple language compiler, as even if our machine doesn't have any compiler to run the program then also we can get the output through cloud.

Keywords: Online Compiler, Collaborative writing, Cloud, IDE.

I. INTRODUCTION

Cloud computing is nothing but an Internet based computing which is used for utilizing hardware and software resources. It allows users to access and share information from devices having internet connection. Cloud computing caters to dynamism, abstraction and resource sharing. Dynamism deals with the fluctuating demands based on seasonal traffic burst, world or regional economy, etc. For example, during tournaments like the IPL, many users visit certain websites for the latest scores, results, fixtures, etc. New severs may be required to meet these increased demands if these websites are not cloud based. After the end of the tournaments, these servers may become idle due to less traffic. So, a decision about whether to buy servers that will remain idle for most part of the year should be purchased, to meet the traffic bursts has to be made. With cloud computing, this issue is solved by the provider of the cloud itself. The cloud provider takes care of fluctuating demands and the organizations have to pay as per their usage only.

Cloud computing provides abstraction which allows developers to focus on their applications. They do not need to worry about the software platform, operating system, web security or updates. The developers are able to concentrate on the core competencies.

Resource sharing provides flexibility to share applications and also other network resources like hardware. Cloud computing also provides a flexible architecture where resources can expand and contract easily. Cloud computing allows users from all around the world to access applications, etc without having to download or install it on their own machines. The owners do not need to reproduce the software or ship it out. The organizations only pay for what they use so, money is saved. Cloud computing can provide virtually unlimited storage as opposed to local servers and hard drives. It is also very flexible and provides easy collaboration with partners. The cloud services can be classified into 3 major types- IaaS, PaaS and SaaS.

IaaS (Infrastructure as a Service):- IaaS is the base layer of the cloud stack The IaaS provider supplies the whole cloud infrastructure like the servers, routers. The customers use these resources as a service on an 'as needed' basis.

PaaS (Platform as a Service):- PaaS is the middle layer of the cloud stack. The PaaS provider delivers a platform i.e. a project environment through the Internet for the developers. It can be consumed using the web browser.

SaaS (Software as a Service):- SaaS is the top most layer of the cloud stack. It is directly consumed by the end user. The only requirement is a web browser and an Internet connection

II. SYSTEM MODEL

The proposed conceptual architecture is a three –tier architecture. The First tier is the Administrator of cloud based code studio, which provide basic functionality i.e. management of our overall profile. Administrator can add the problem definition / Aim of project and assigns it to client. The second tier is nothing but this assigned client. This client should be registered and must log in to the system with authentication process so as to perform successful operations.

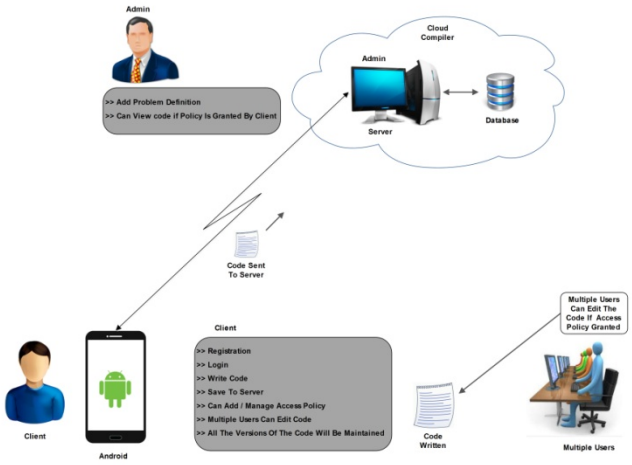

Fig.1: Collaborative writing with online compiler

Authenticated Client have the policy to manage the access policy, means he can add any number of people to edit the code present in last i.e. third tier. Client provides selected users to edit i.e. write policy and others only to read or could restrict them from both. Third tier consists of users who edit the code if policy is granted. They save or send the final copy to client or server. So all these versions of code will be maintained at client level. As we are using online compiler for languages like java, c, c++ there is no need of installing compilers on host PC. Then the client will determine the best outcome from the users and send back to the administrator. As the document is going to be centrally located on cloud access to these client and users will be easy. As system provides the functionality to edit it collaboratively the time required for modification, updating will be much more less than usual manual systems.

III. Previous Work

During the development of any tool it is necessary to determine the time factor, economy and company strength. For determining which operating system and language can be used for developing the tool, all above conditions should be satisfied. When the programmers start building the tool, he needs lot of help and support. It can be achieved from senior programmers, from book or from websites. So before building the system the above consideration are taken into account for developing the proposed system.

The author Mayank Patel has proposed the system only for online Java compiler [1]. Second author has done the similar kind work for C, C++ [3]. But there is lack of collaborative approach in both systems. The cost required for installation of each platform according to language was very costly and time consuming. It was also very difficult to know which host PC have or have not the particular platform. As for interpreters takes much time in runtime performance and JIT takes a lot longer to develop. Also the software like VMware, it uses an instruction set that also happens to be used on real hardware. It is the same basic concept as a language VM, in which it pretends to be a machine that is not actually present, but in practice it is different and very complicated.

The current system does not provide a web application to run collaborative cloud based compiling tools on mobile devices. With the help of references of Mobile-Cloud-based-Compiler-A-Novel-Framework [4] we can implement the above idea. The main advantages of creating your own instruction set are to lower the long compile/optimization times and slow interpreters. When we compile a Java class, we don't have to do register allocation or any of that traditional compiler stuff. But only for the parts of code that we run much times, and spread out over the run of the program. The java virtual machines' instruction set is same to Java that the initial compile is quick, and it is simple and easy to read unlike Java source code [1].

In previous system, users were restricted to access the same file at the same time. This would lead to user's dissatisfaction. Also indirectly could face the problem of time management. To select the correct versions of document file was the most tedious job.

IV. Proposed Methodology

On the concepts based on paper [1] and paper [2], and taking the theme of that paper we propose a system model illustrating the complete flow of our system as shown in Fig 2.

Integrated Development Environment (IDE):

An Integrated Development Environment is a program for software developer that combines the functions of a text editor, an interpreter or a compiler and run time facility to simplify coding and debugging. In earlier times, a programmer had to edit the files, save them, run the compiler and the linker then build the application and run it through a debugger. Today, the IDE's bring the editor, compiler, linker and debugger to one place to increase programmer productivity. They also have support for certain servers like the GlassFish Server, Oracle Weblogic Server, Apache Tomcat, JBoss, etc. IDEs also provide project management tools.

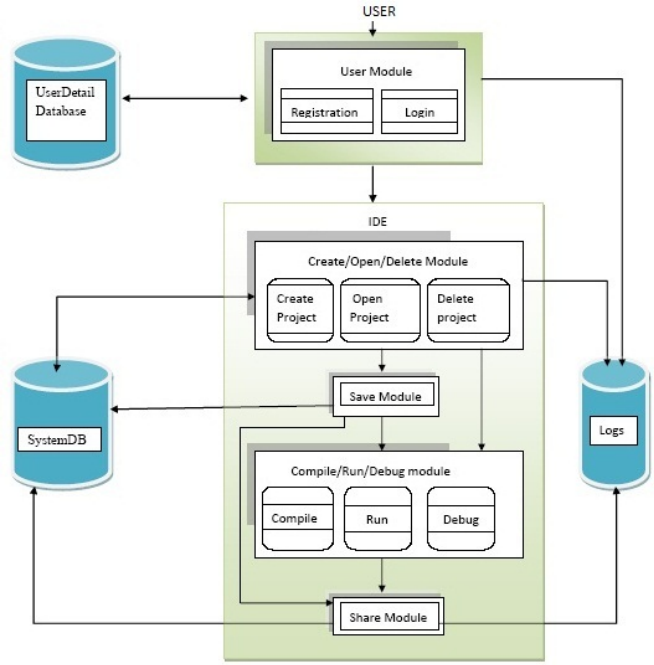

Fig.2: Architectural Model

The main features of an IDE are:

1. Syntax Highlighting:
   To highlight different parts of the program Colors are assigned.
2. Automatic Editing:
   Programmers can format their programs so that blocks of code are indented together.
3. Automatic Code Completion:
   An IDE provides the programmer with a list of possible options while coding.
4. Access Database:
   IDEs can help applications access different databases and query data within them.
5. Compilation Execution And Debugging :
   IDEs have the ability to compile and execute programs. The programmers do not have to switch from the text editor to a terminal window.
6. GUI Builder:
   GUI can be created by dragging and dropping components onto the canvas.
7. Optimization:
   IDEs have profilers built into them to highlight the areas where the code can be improved.
8. Version Control:
   The previous versions of the source code are stored and managed by the IDEs. Some examples of well known IDE's are Netbeans7.0, Eclipse IDE for Java Developers etc

Browser Based IDE's (BBIDE) to Code in the Cloud:

Many a times, people would want to write programs on machines that do not have required software. If, for example, they want to code a java program on a device that does not have an IDE or a JDK, they will have to download hundreds of megabytes followed by a lengthy installation process. This can be very inconvenient. Browser Based IDE's help overcome this drawback. Browser Based IDE's are Integrated Development Environments that are accessible to everyone through a web browser and an Internet connection. It is basically elevating the coding platform to an online environment where the OS issues are eliminated and hardware independence is achieved. It provides increased portability and accessibility. It can be used as a programming environment for multiple people. Browser Based IDE is Software as a Service (SaaS). The IDE is one of the applications on the top most layers of the cloud stack. It is directly consumed by the users.

The advantages of Browser Based IDE's are:

1. Increased portability and accessibility
2. Operating System and hardware platform independent
3. Provides with the ability to program with devices having Internet connection
4. Easy pair programming, sharing and collaborations from different locations

This project creates Integrated Development Environment for the multiple languages to code, compiling, run, and test and debugging the code using the Internet and a web browser. On the Cloud accessible from various devices, the IDE will permit easy development, testing along with debugging of applications. The devices that have an Internet connection and a web browser will have access to the IDE present in the cloud. The IDE will be hardware independent, and all the operating system issues will be eliminated. This IDE can be used instead of or alongside the existing desktop IDE. It will also have support for uploading the existing code in order to test it in the cloud or for sharing with peers. It will also enable easy compilation, packaging and deployment of the code in the cloud.

V. CONCLUSION

This project aims at creating a Browser Based IDE to code Java and the other language in the cloud and also aims to

provide the special feature of real time collaboration for the users. The modules and design for this project have been conceptualized and the hardware and software requirements for development and usage have been analyzed and documented successfully.

So our system provide an interactive and user convenient tool to run user program online without installing any plug-in and software on own PC. This help to run code from any remote location. Also the system helps to provide access to user to writes data into file at same time from different location that's makes good user convenient feature. A key design aspect was the use of cloud computing writing tools and their APIs to build tools that make it seamless for students to write collaboratively either synchronously or asynchronously.

VI. Future Scopes

The system software can be extended in the future to include Java Platform Enterprise Edition technologies like JSP, Servlets along with other advanced functionalities like syntax highlighting, code compilation, sharing code with hyperlinks and support for other languages. System provide a feature to change file format online instead use tool or software to change file format after file downloading so make user efficient feature to change the file format at the time of downloading..

References

[1] Mayank Patel, "Online Java Compiler Using cloud computing", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-2, January, 2013.

[2] Rafael A. Calvo, Senior member, IEEE, Stephan T. O'Roureke, Janet Jones, Kalina Yacef, and Peter Reimann, "Collaborative writing support tools on the cloud", IEEE transactions on learning technologies, vol. 4, no. 1, January-March 2013.

[3] Aamir Nizam Ansari, Siddharth Patil, Arundhati Navada, Aditya Peshave, Venkatesh Borole, "Online C C++ Compiler using Cloud Computing", IEEE, May 2011.

[4] Mahendra Mehra, Kailas K. Devadkar, Dhananjay Kalbande,"Mobile-Cloud-based-Compiler-A-Novel-Framework-For-Academia Paper ", International Journal of Advancements in Research & Technology, Volume 2, Issue4, April - 2013.

[5] A.Rabiyathul Basariya, K.Tamil Selvi "Centralized C# Compiler Using Cloud Computing" International Journal of Communications and Engineering Volume 06– No.6, Issue: 02 March2012.

[6] www.penflip.com