Improved Interaction in Web-Based Cloud IDE

Rajesh Hegde, Karunakara K

Abstract— Internet has added more dynamism to this fast changing world. Most of the time, people wish to use internet to solve their problems. As of now, most of the events happen via on-line. When a new program/project is developed, developer may wish to test his program/project to know its performance without actually investing on the compilers and other libraries but to use cost effective solutions before being developed as a separate software package. If such solution is made available on-line at a learner's cost it may be attract entrepreneurs and newbies to develop new ideas and software solutions. For compiling and executing the source program, user needs a compiler package tools. The overhead of installing compiler to each machine can be avoided by using on-line compilers. A web-based environment has been developed for learners/developers to write programs in different languages. The proposed system runs on a Linux environment and provides each user with separate subdirectories that is completely isolated from others. The main aim of designing of this system is to ensure easy way of program development and to use on-line facilities to execute and debug the program. The client machine doesn't need a compiler kit. The proposed system has better interaction with the user by accepting the input at run time, process it and produce the result. This web-based application can be used across any network and platform. It also eliminates the hassle of installing the compiler on to each computer. There by reducing the cost factor during project development.

Index Terms—Cloud Computing, Web-Based compilers, Interactive compilers, Cloud IDE.

I. INTRODUCTION

Internet has become integral part of the human life. It is the desire of the users to use Internet to maximize its benefit to solve their problems. For instance, a program developer may wish to solve a problem by writing a program by selecting a programming language. It depends on the nature of the problem as each language has its own merits. That is user may prefer to use C language to solve the problem which are mathematical intensive, Java for web applications. To work with multiple languages he is expected to install different compilers on his machine. This installation steps is different for each language. Further, updating the compiler kit has also an overhead. These native compilers occupy user's disk space. To overcome these problems and to ensure the users spends most of their times for developing the desired solutions to the problems, a system has been proposed to solve the required problems through the cloud services. A new era of Internet has been started with the evolution of Cloud Computing. As the Google defines Cloud Computing is a practice of using a network of remote servers hosted on the network to store, share, process and manage data rather than a own local server.

Use of Cloud Computing services will reduce the overhead on the end-user, increase the flexibility, reduce total cost of ownership and provides on-demand services apart from other advantages. This technique does not require the end-user to know the physical location and configuration of the system that provides services to him. 3 service models have been identified with cloud computing techniques.

1. IaaS - Infrastructure as a Service: This will provide the physical or virtual infrastructure to the end user which can be reallocated any number of times on-demand. Typical start-up companies may not have enough capital to buy their own servers. IaaS solves this issue by providing them infrastructure on subscription. The consumer is provided the capability to have resources like CPU cycles, persistent storage, high speed secure network etc. These resources will be used to deploy and run the software such as operating system and applications. End user does not need to manage or control the underlying technology which is providing him scalable services.
    Ex: Amazon EC2 [1].

2. PaaS - Platform as a Service: This is a service which will provide a platform for the consumer on which they can build and deploy their services and applications. The customer is provided with infrastructure, platform and programming language tools to develop applications. Platform is the extra layer that is managed by the cloud vendor. User will have the control over the deployed applications and possibly application hosting environment configuration.
    Ex: Google App Engine [2].

3. SaaS - Software as a Service: This model will provide on-demand web based applications to consumers. Vendor will host the applications and are responsible for running the application. Consumer needs only a subscription to get service. The end user is provided the ability to use the vendor's applications running on a cloud infrastructure to perform their work on the fly. The applications can be accessed from diverse clients such as smart phones, tablets, PC etc. Hosted application is also managed by the cloud vendors except some configurations can be changed.
    Ex: Google Docs [3].

In this paper, a model is proposed that provides compile/Execute/Debug services to the end-user using SaaS model. This is a Web based application which can be accessed across the globe and through any devices like smart phones, laptops, desktops, etc. that have a browser, an Internet connection as shown in Fig. 1 The proposed model is implemented using open source tools and applications.

Figure 1. Cloud Devices

The online capability of the proposed system to compile and execute the code of different language will reduce the overhead of CPU cycles and disk space of the client machine.

II. RELATED WORKS

Many web-based IDEs are released during the past few years. Their type varies from simple programming language tutorials to complete solutions for developing and deploying web applications. A few existing browser based coding environments are Cloud9 IDE, CodeRun Studio, ideone, Eclipse Orion, codepad etc.

1. *Cloud9 IDE* [9]
   Cloud9 IDE supports HTML, CSS, JavaScript, etc. It is mainly for web development. It has support for real time collaboration.
2. Eclipse Orion [10]
   Eclipse Orion is mainly for web development and it supports HTML and JavaScript. Both Cloud9 and Orion IDEs do not have support for C language.
3. Ideone [11]
   Ideone is not an IDE. It is like a pastebin that supports compilation and debugging of code in many languages including C, python. But it does not permit creation of projects and does not provide interactivity to user at run time.
4. Codepad [12]
   Codepad is also similar to Ideone. But it does not provide the facility to give input to program.

Sravanthi Emani [4], N.B. Pokale, Arti Chetwani and Archana Patwari has discussed about the Web Based C IDE and they address dynamic Input to the program code issue as a challenge. They have provided an Architecture of IDE, but this architecture suffers from interactivity. Max Goldman [5], Greg Little, and Robert C. Miller has also mentioned that they are trying to improve for console input, and supporting languages other than Java. Lakshmi M. Gadhikar [6], Lavanya Mohanv, Megha Chaudhari, Pratik Sawant and Yogesh Bhusara have discussed "Browser based ide to code in the cloud" but they found dynamic input issues is a challenge. Mayank Patel [7] also presented "Online Java Compiler Using Cloud Computing" but he didn't mention about taking the input at run time. Timo Aho [8], Adnan Ashraf, Marc Englund, Joni Katajam¨aki, Johannes

Koskinen,Janne Lautam¨aki, Antti Nieminen, Ivan Porres, Ilkka Turunen have presented "Designing IDE as a Service".

III. STATE OF THE ART

At present many cloud service providers are offering on-line compiler as a service. Using these facilities user can execute their programs but they are less interactive with end-user causing great inconvenience to user to test their solutions by inputting the data dynamically. For the existence type of services the data need to be supported at the time of compilation. This denies the programmer need to test his solution for verities of data under different scenarios.

```
C, pasted just now:
1  #include <stdio.h>
2  int main()
3  {
4      char name[20];
5      printf("Enter name : \n");
6      scanf("%s", name);
7      printf("Hello %s !!!", name);
8      return 0;
9  }
```

```
Output:
No errors or program output.
```

Figure 2. Codepad Error

```
edit    fork   download
1.   name = raw_input("Enter name")
2.   print name

Runtime error

stdin
Standard input is empty

stdout
Enter name
stderr
Traceback (most recent call last):
  File "prog.py", line 1, in <module>
EOFError: EOF when reading a line
```
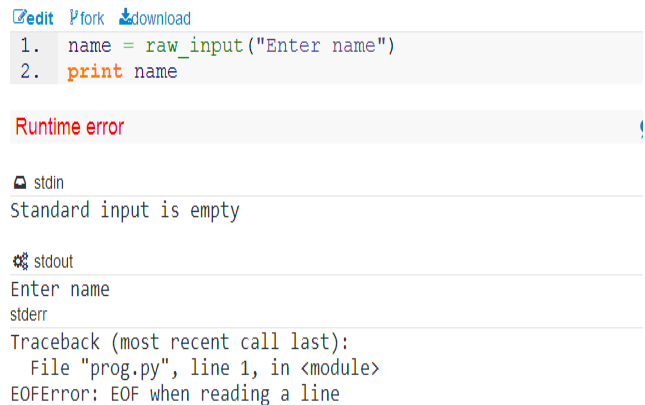
Figure 3. Ideone Error

The existing on-line compiler such as codepad and ideone have limited or no facilities to have user interaction at the execution time. This means each time if the programmer needs to be checked different data, it needs to be changed at the source level. This further requires recompilation and execution. The snapshot of codepad the ideone on-line compiler environment is shown in Fig. 2 and Fig. 2 respectively. Fig. 2 shows scanf is not able to read value for 'name' variable and no output is shown. Fig. 3 shows 'raw_input ()' function is not able to get input from user, throws Runtime error. This problem is not only bound to codepad and Ideone, many online compilers suffer from this problem [13][14][15][16]. To make on-line compilers more useful and provide interaction for developer it is necessary to enhance the capability of the on-line compiler.

This also helps software testing phase to check the behavior of the program under different scenarios. This provision of providing the input at the execution time will help in separating the development with testing phase. This gives an opportunity to hide the code if needed, during testing phase which in turn ensures security of the source code being developed.

A few on-line compilers provide facility to enter standard input as shown in Fig. 4. However this doesn't solve the problem completely as inputs are static in nature. Users are expected to give the input before program being run. Each time the data is changed the program need to be recompiled and run. This can be feasible for small programs with limited interaction.

Figure 4. Ideone Interface

Consider a situation in which user has to make choice at every step. It will become then difficult to specify the input before execution of the program. It is not possible to take dynamic decisions. For the program shown in Fig. 5, it is not convenient to give series of inputs before running the program.

Figure 5. Classic Problem

IV. IMPROVED METHOD

However the earlier systems were able to get the input from the user before actually executing the program. Proposed system explicates the interactive way of executing the program. With the advancement in the technology, users are no more interested in refreshing web pages, rather they demand an asynchronous communication which interact with the server without refreshing page. The suggested system will

make an asynchronous connection while compiling and executing the code and the connection will be alive as long as the program under execution is exited. This asynchronous connection will be responsible for sending the data back and forth. Considering the drawbacks of existing methods a solution is proposed here which will enable the compiler to accept the input from the user at run time asynchronously. A snapshot in Fig. 6 shows that program is accepting the input from the user interactively. This will give a feeling that the program is being executed in the client machine itself. The proposed system can also be used to offer Interpreter as a Service to the user. Interpreter expects much interaction from the user than that of compilers. Proposed solution can also be used to solve the issues with interpreter by offering asynchronous connection between server and the client.

Figure 6 Running C Program

V. IMPLEMENTATION

The system can be accessed from anywhere by connecting it via Internet. On the Linux kernel, an interface is developed to enable the compilation of the programs. In case of compilation errors, the details of the error will be displayed to the user to help him correct the code accordingly. The system will include an interface between the compiler and the web browser. The code entered on the browser will be directed to the compiler and the corresponding results of code execution will be shown on the browser. If any inputs to the programs are needed then it can be entered interactively. Architecture of proposed System is given in Fig. 7. For the implementation of Web based Compiler, an open source IDE Codiad[17] has been used. This is a Cloud Application and it can be deployed on any Linux servers. For example it can be deployed on Amazon EC2 running Ubuntu 12.04 virtual machine. IDE front end will look like as in Fig. 8 it will internally compile, execute and then produce the output. If any errors found, it will be displayed in Output Console. If it is error free then Output Console will show program output/result.
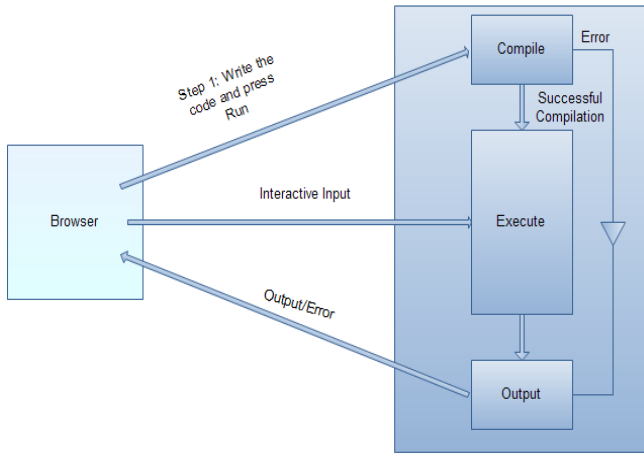
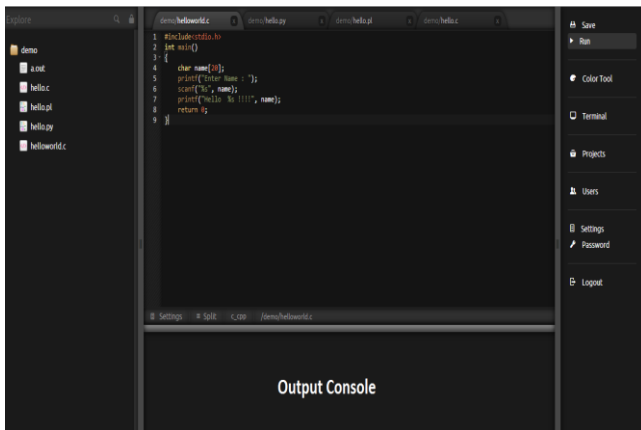**Figure 7. Architecture**

Figure 8. IDE Interface

When the user presses "Run" button, the following steps are carried out.

1. Identify the programming language.
2. Build appropriate command for compiling and executing the program.
3. Generate unique id to represent the session.
4. Send asynchronous request to server with the commands.
5. Execute the command.
6. Create buffer file for input and output.
7. After command execution finished, close the connection, flush the buffer.

VI. EXPERIMENTAL RESULTS

For demonstration purpose the application is hosted on Ubuntu 12.04 LTS using VMware Player. The machine used with the following configuration
Server:
   Linux Operating System (Ubuntu 12.04 LTS)
   1GB RAM
   2.4GHz CPU
Client/User:
   PC with Chrome Browser
IDE is implemented using HTML5 technology which avoids installation of additional plug-ins. The system proposed here has ability to execute many language codes using same interface. For demonstration C and Python language compiler services are implemented. C program execution is shown in Fig. 6. Fig. 9 shows python program execution. Debugging facility is also integrated within the system. It will

find the compile time errors. If any error occurs during compilation then it will be displayed on Output Console as shown in Fig. 10.
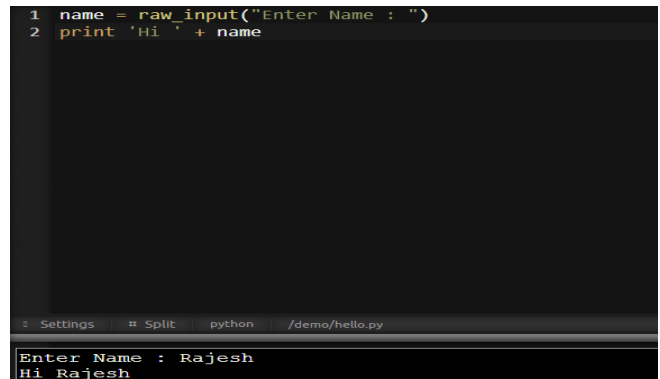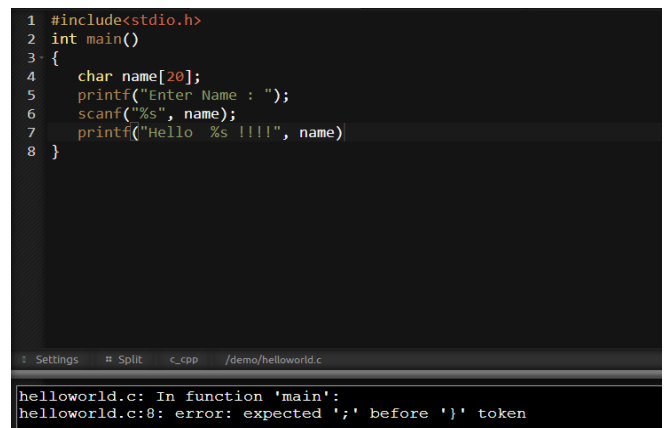

Figure 9. Python Program Running

Figure 10. C Program Debug: Shows the Error Debugging [Semicolon is Missed in Line 7]

The future of the system lies in using these services through portable devices without the need to install heavy software on the devices. A powerful browser is enough to support use such apps satisfactorily on Tablets and smart phones. Thus, there would be no need of labs and large infrastructure in the future, a Tablet is all that is needed for the students to execute their programs. While executing the programs, server side CPU cycles are used. Clients CPU cycles are available for other work. So it will lead to evolution of thin clients. Demonstration of CPU usage in client machine is shown in below figures Fig. 11 Fig. 12 and Fig. 13. Fig. 11 snapshot is taken before executing the CPU intensive code in the client machine. Then code is executed in the client's machine using native compiler in command line. Fig. 12 shows the CPU usage of the above case. Then the same code is executed using the proposed system. As it takes CPU cycles of the server to execute the code client's machine's CPU is not consumed. Observe the CPU usage in Fig. 13. It is nearly equal to that of usage before. It concludes that proposed system has a great effect on the CPU usage.
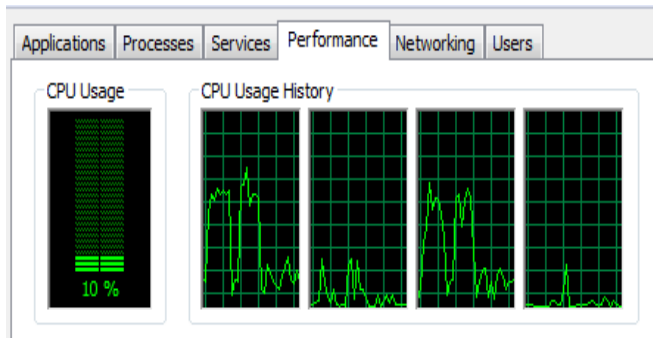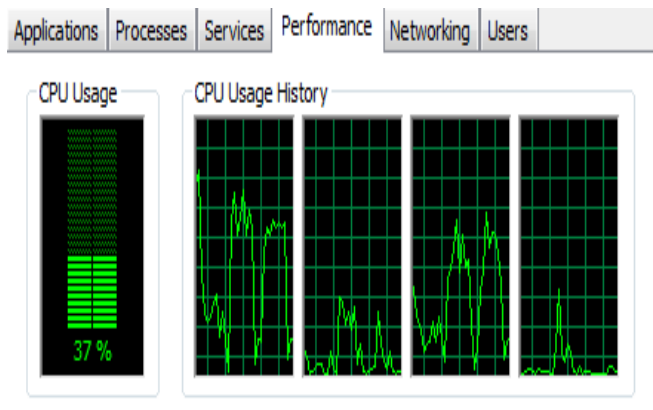
Figure 11. Before Running
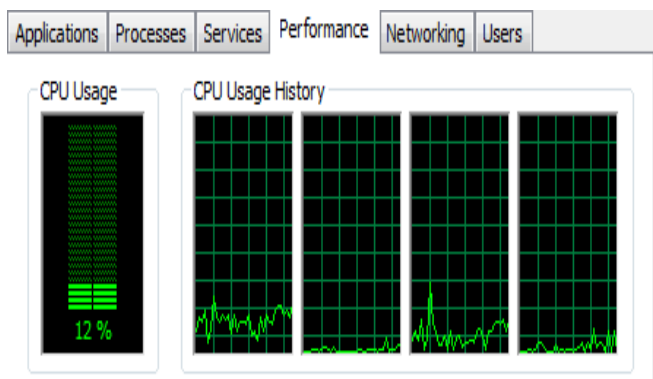

Figure 12. Run on Local System

Figure 13. Run Using IDE

VII. CONCLUSION

We can conclude that this system is more powerful and highly advantageous to users. This system ensures the execution of the programs on any platform and avoids the configuration of the system in running programs. Also the user doesn't need to install the compiler packages on every machine. This saves a lot of time and improves efficiency. Thus he can use his precious time on developing and testing the code. The system can be easily get accessed from anywhere. It is comfortable to understand and use without any complicated commands. Also the memory space in the machine is saved as the programs and the included components do not occupy any space on the client's machine. As this system is designed with interaction with the user at run time, user can feel more live in executing the program. Different language programs can be coded and executed using same interface which will fulfill his requirements.

REFERENCES

1. Amazon, Aws | amazon elastic compute cloud (ec2) - scalable cloud hosting (2014). URL https://aws.amazon.com/ec2/
2. Google, Google app engine (2014). URL https://appengine.google.com
3. Google, Google docs - online documents, spreadsheets, presentations (2014). URL https://docs.google.com
4. S. Emani, N. Pokale, A. Chetwani, A. Patwari, Web based'c'ide: Approach., International Journal on Computer Science & Engineering 4 (3).
5. M. Goldman, G. Little, R. C. Miller, Real-time collaborative coding in a web ide, in: Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM, 2011, pp. 155-164.
6. L. M. Gadhikar, L. Mohan, M. Chaudhari, P. Sawant, Y. Bhusara, Browser based ide to code in the cloud, New Paradigms in Internet Computing (2013) 59-69.
7. M. Patel, Online java compiler using cloud computing, International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN 2278-3075.
8. T. Aho, A. Ashraf, M. Englund, J. Katajam• aki, J. Koskinen, J. Lautam• aki, A. Nieminen, I. Porres, I. Turunen, Designing ide as a service, Communications of Cloud Software 1 (1).
9. Cloud9, Cloud9 ide | your code anywhere, anytime (2014). URL https://c9.io/
10. Eclipse, Orion (2014). URL http://www.eclipse.org/orion
11. Ideone, Ideone.com - online compiler and ide c/c++, java, php, python, perl and 40+ other compilers and interpreters (2014). URL http://ideone.com
12. Codepad, Codepad (2014). URL http://codepad.org
13. HackerEarth, Online compiler/interpreter | | codetable (hackerearth) (2014). URL http://code.hackerearth.com/ 255
14. compileonline, Compile and execute c online (gnu gcc version 4.8.1 (2014). URL http://www.compileonline.com/compile_c_online.php
15. py-ide online, Py i/o - best python ide online (2014). URL http://py-ide-online.appspot.com
16. CodeTwist, Codetwist : Simple c, c++ and java online interpreter and compiler (2014). URL http://codetwist.com
17. Fluidbyte, Codiad web based ide by fluidbyte (2014). URL http://codiad.com