**Cloud Based Coding Environment Based on Linux LXD**

**Abstract**

Everyday, new technologies emerge in the field of Information Technology. One such technology is “Cloud Computing” which has been adopted by a number of large-scale enterprises to make their work more efficient. But why are companies and enterprises moving to cloud? The answer lies within the scalability, portability, efficiency and the phenomenal capabilities of the Cloud Computing.

Although the practice of transferring data and providing services over a cloud platform is prevalent now, the idea of Cloud Based Coding Environment is still uncommon, especially in educational institutes that still rely on desktop IDEs.

This paper, discusses the problems which the programmers/students face when coding on a desktop IDE and how a cloud-based coding environment can be used as a powerful alternative. This paper also presents a working prototype which was built as a part of research to understand the full potential of a cloud-based IDE and understand the compatibility level of the end-user when asked to shift from a conventional IDE to cloud IDE.

Keywords: Linux LXD, Linux Container, Desktop IDE, Cloud IDE

1. Introduction

In todays era, most of the things we use are based on cloud now. For instance, listening to a song on “SoundCloud” and “iTunes”, uploading pictures on “Instagram” or transferring data to “DropBox”, all makes use of cloud platform. Amazon Web Services is another example of a cloud platform where businesses can create large-scale application without relying on physical data centers or storage areas. Our team studied the need for a cloud-based IDE to help students learning software development, program and code conveniently while not worrying about additional hardware resources. Additionally, as a part of an educational institute, the same platform can also provide instructors the capability to evaluate the student’s code while students will be able to collaborate with their peers as well.

**Literature Review**

In our research, we found out that the existing available systems such as AWS Cloud 9, SourceKit, Codeanywhere provide a full-fledge cloud IDE solution. However, they are all paid services. Whereas, when we searched other solutions presented in different papers, our findings revealed various issues with the proposed solutions.

A solution demonstrated in one paper (Cloud on Compiler) proposed a similar prototype which made use of the following technologies .NET Framework and MySQL. The system uses two databases: one for storing user info and other for storing user files. This could be made more efficient by using just one database to manage aspect of the program.

Another solution by the name, “Cloud Based Program Evaluation Using Online Compilers” provides a platform to educational institutes for conducting exams and evaluate user submitted code on the cloud. However, the system does not provide any functionality to create exams on the cloud as well.

**Methods**

**Participants:** The target audience for our research activity were students doing majors in Software Development as well as their instructors and tutors who were also professional developers. The questionnaire was presented to 90 people who were divided into two groups.

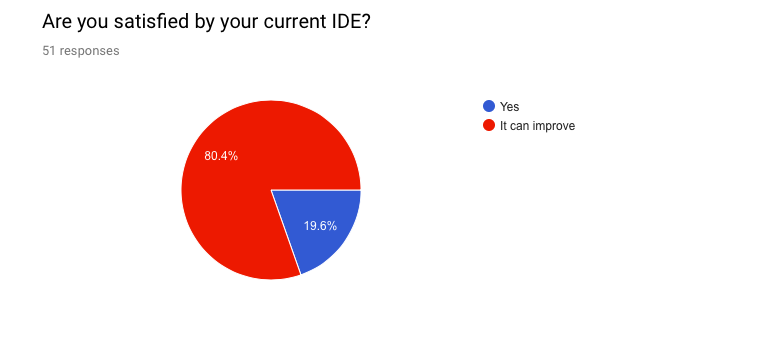
**Group A:** Students

**Group B:** Instructor

**Research Activities:** The data for this research was collected primarily through experiments, survey and questionnaires. Interviews were also conducted with the leading faculty members of different universities.

**What Challenges Does Desktop IDE Serve?**

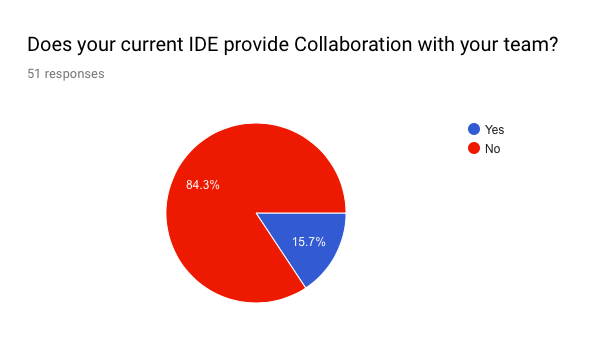
To understand the problems faced by students when coding on a desktop IDE, we interviewed them and presented a questionnaire. 80.4% of the response indicate that students believe their current IDE can be improved while 19.6% are satisfied with the current IDE.



**Collaboration**

One of the major drawback of Desktop IDE is that it does not provide any functionality for different users to collaborate. For instance, if User A and User B are working together on the same project and User A is done with his/her part of the work. To enable the User B to continue working on the same project, User A would have to save and send the file to User B. User B will then download and export all the setting on his/her desktop IDE in order to continue working. Our survey indicates that 84.3% of the users are using a desktop IDE which does not have a function for collaboration.

This problem is addressed in our proposed solution which provides the users to save their working environment on the web and provide anyone the access to view and make changes accordingly.



**Coding on the Go (Mobility)**

Mobility or Coding-On-The-Go is a term often used to describe programming remotely from any location and using any computer. Desktop IDEs do not support mobility as they require installation on every new computer a user wishes to work on. Cloud Based IDE does support mobility as it does not require any installation and can be accessed on any computer through internet connectivity.

**Security Concern**

Using a desktop IDE means a user would be connecting different portable hard disks or USB devices for various purposes. This increases a risk of your computer getting infected with a virus and can cause data loss as well.

**Analysis of the Proposed Cloud Based Solution**

Studying the problems faced with current Desktop IDEs and some of the existing cloud based IDEs, we developed a solution which tackles all these issues and offers a much enhanced and powerful platform to the students where they can code, collaborate and have their work evaluated by the field professionals. In the proposed project, we currently implemented four compilers including C, C++, Java and PHP. Emphasis was put on evaluation process as well where educational institutes can conduct examination through the proposed system and evaluate student’s performance.

**Architecture**

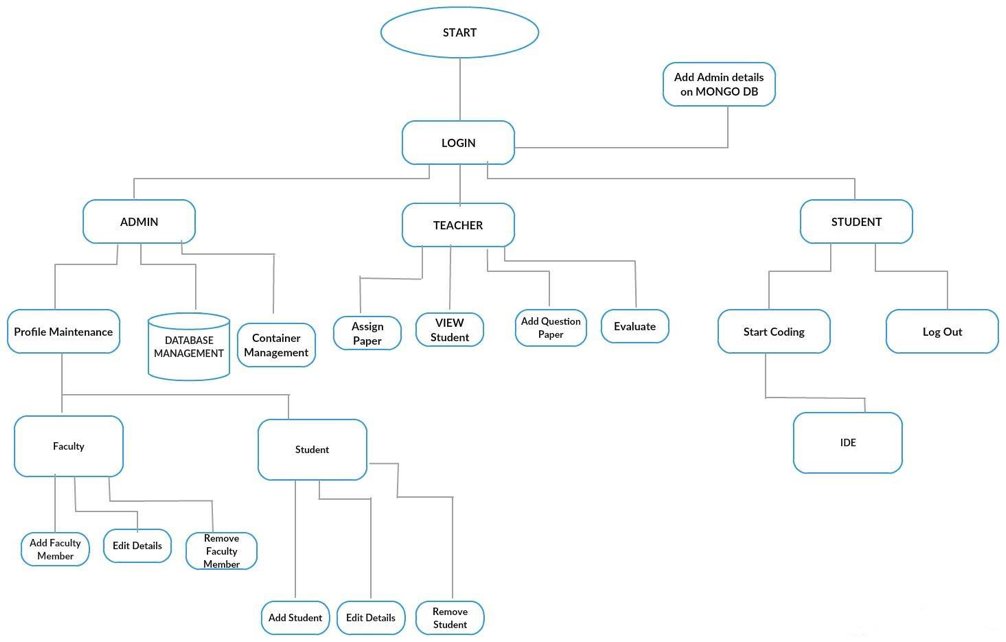
The detailed working of our proposed cloud based coding solution is shown in the figure below. The project starts with a login screen where the user is asked to input his/her credentials. The details are matched against the database (MONGODB) and the user gains access to the system.

Depending on the user type, access will be provided to one of the following panels, Student, Admin or Admin. The Admin is responsible for managing the system, adding users and creating a programming instance for each user.

The student can be any regular user who wishes to program using the cloud based solution or is entitled to give a certain exam.

The Faculty member is a professional who can code as well as evaluate other submitted programs.

The Student Panel provides the students capability to program on any environment they wish. Moreover, students can save and view they program at any later time they wish.

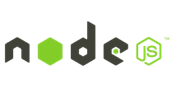


**Key Technologies Used:**

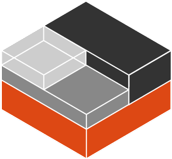
Following are the key technologies used in our proposed solution:

** MongoDB**

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas

** NodeJS**

**Node**.**js** is a platform built on Chrome's **JavaScript** runtime for easily building fast and scalable network applications. **Node**.**js** uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

** Linux Container LXD:**

Linux containers, in short, contain applications in a way that keep them isolated from the host system that they run on. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. And they are designed to make it easier to provide a consistent experience as developers and system administrators move code from development environments into production in a fast and replicable way.

LXD is a next generation system container manager. It offers a user experience similar to virtual machines but using Linux containers instead.

**MVC Web Application Framework**

In terms of choosing the web application framework, we could have opted for ASP.NET Web Forms but our choice was MVC due to the clear-cut separation between Model, View, and Controller layers. Both frameworks have their pros and cons for example:

**MVC Pros:**

1. Enables the full control over the rendered HTML.
2. Provides clean separation of concerns(SoC).
3. Enables Test Driven Development (TDD).
4. Easy integration with JavaScript frameworks.

**WEB Forms Pros:**

1. It provides RAD development
2. Easy development model for developers those coming from winform development.

In terms of Database, MySQL was an alternative but due to the fact that MongoDB is schema-less. Moreover, MongoDB enables to build applications faster, handle highly diverse data types, and manage applications more efficiently at scale.

**Key Modules:**

**Start Coding:** This Component opens up the IDE assigned by the administrator or the Instructor.

**Student Panel:** This component provides student with options including “Start Coding”, “View Performance”, “Take Exam” and a “Logout”.

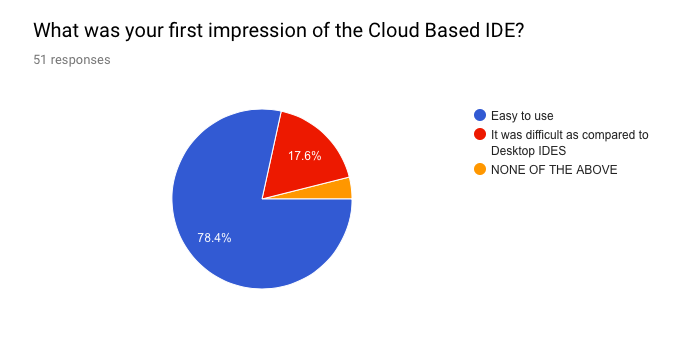
**View Performance:** This component allows students to check the grade awarded by faculty member.

**Faculty Panel:** The component allows the instructor to create and assign exam to the students. The faculty can also evaluate the code submitted by the students and grade them accordingly.

**Create Exam:** This component allows Faculty members to create an exam for the courses they have been assigned.

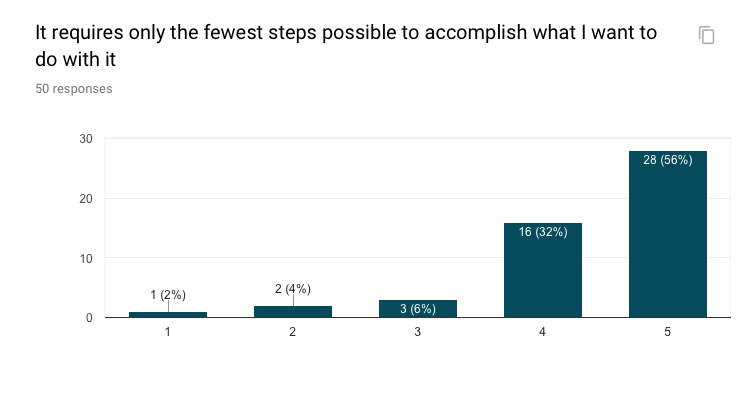
**Results:**

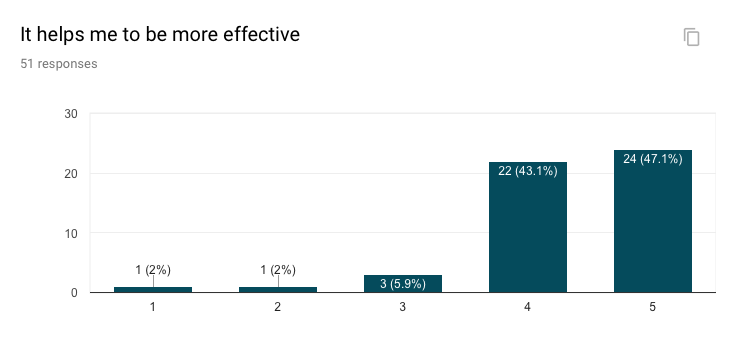
Another survey was conducted after presenting the proposed solution to the users in order to find out whether or not they feel comfortable using the cloud based IDE. Research indicates that the initial impression of the majority of users was that the Cloud Based IDE was “Easy to Use”.



**Usability Testing**

Our observation concludes that users found the cloud based IDE to be better in terms of usability compared to a desktop IDE. For new users, setting up a desktop IDE such as “Visual Studio” can be daunting task as it requires a number of preliminary steps to start and execute project. On the other hand, the Cloud Based IDE provides the coding platform with just a click of a button. During the post-experiment survey, participants were asked how efficiently the proposed cloud based solution helps them accomplish their task and the result can be seen below. 56% of the participants rated 5/5 for software efficiency and effectiveness.





**Performance and Speed**

To understand and compare the performance and speed of desktop IDE to our cloud based IDE, we studied the system requirements of some popular desktop IDEs.

**Hardware Requirements for Visual Studio Vs Cloud Based IDE**

|  |  |
| --- | --- |
| * 1.8 GHz or faster processor. Dual-core or better recommended * 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine) * Hard disk space: up to 130 GB of available space, depending on features installed; typical installations require 20-50 GB of free space. * Hard disk speed: to improve performance, install Windows and Visual Studio on a solid state drive (SSD). * Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher. | * A computer system with an operating system such as Windows 7, Linux Distribution, OSX 10.5, Ubuntu or higher. * Internet Connection |

**Installation Time**

Our experiment result indicated that a Desktop IDE such as Visual Studio 2017 takes about 40 to 45 minutes for installation. A software such as Android Studio takes even more time to install on a regular computer. In comparison, a Cloud Based IDE would require just an internet connection and absolutely no setup time.

**Security and Protection**

As mentioned previously, using a desktop IDE can lead to a computer getting infected with a virus or malware. In other solutions we studied, the security concern is genuine as a small virus on the server end can cause serious disruption to the whole platform. However, our proposed solution makes use of Linux Containers and each user is assigned an individual container for their work. In case of a virus, it will effect only the specific container and not the whole system, making our cloud based solution secured.

**Conclusion**

Through our thorough research, we conclude that the majority of the participants with whom we tested our proposed system, shown positive response in terms of usability, performance, mobility and other aspects as well.