

Smart Cloud Com

Debasish Pradhan, Sandeep Sahoo

Gandhi Institute of Excellent Technocrats, Bhubaneswar, India

Suddhananda Engineering & Research Centre, Bhubaneswar, Odisha, India

ABSTRACT-Today, internet has been widely used. With the use of internet, all things are being done online. Thus creation of online compiler is the need of the hour. The main aim of this paper is to provide online compiler which helps the user to write the program easily, compile it and debug it online. The paper proposes to describe centralized compiler which reduces the problems of time, cost and storage space by making use of the concept of "Cloud computing".

Cloud Computing is a model for providing computation, software, data access and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. The present paper on "Smart Cloud Com" uses CodeDom (Code Document Object Model) Technology to generate Centralized C# Compiler. The CodeDom allows program to be dynamically created, compiled, and executed at runtime. The CodeDom provides a language independent object model for representing the structure of source code in memory. The System.CodeDom namespace contains number of classes, interfaces, and enumerations. The user(s) paste/write the code in the main window and compile it and execute the code. By using smartcloudcom we can conduct online practical examination too.

Key Words: Compiler, Cloud, Centralized, CodeDom

I. INTRODUCTION

Cloud Computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. This project aims to create an online compiler i.e., c# compiler, which helps to reduce the problems of portability of storage and space by making use of the concept of cloud computing. The ability to use different compilers allows the programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover, a web based application can be used remotely through any network connection which is platform independent. The errors/output of the compiled program can be stored in a more convenient way. Also the trouble of installing a compiler on each computer is avoided. Thus these advantages make this application ideal for conducting online examinations.

Cloud based compiler mainly deals with providing a platform to compile and execute programs that is not dependent on any platform related restriction or compilation. Compilers run the programs and convert

them into an executable format from a text format. A compiler which has to be installed manually on every system needs a lot of space. A program becomes platform dependent. Once it is compiled, it is difficult to carry the same program code to multiple systems if a single system is used. There is no local installation or maintenance work necessary. The software would be provided to the end user using a SaaS cloud.

Cloud Computing is not simply collecting the computer resource, but providing a management mechanism and services for millions of users simultaneously. The users need not care how to buy servers, software, solutions and soon. Users can buy the computing resource through internet according to their own needs. The main reason for creating the project is to provide a centralized compiling scheme. Also, it acts as a centralized repository for all the codes written. The other major advantage that this system will have over the others is that it will make the users system lightweight i.e. there will be no need to maintain separate compilers at the client side. Also, the process of maintenance and distribution of dynamic usernames and passwords will be greatly simplified. In addition to these, it also makes authentication and personalized task distribution possible.

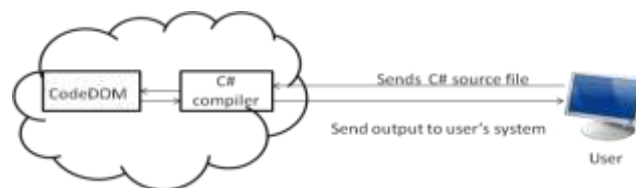


Fig1:Diagram showing structure of data and program components

A compiler, which is the heart of any computing system, transforms source code from a higher level language to a lower, machine level language. This is mainly done in order to create executable files and run to execute the program and its instructions. Cloud based compiler mainly deals with providing a platform to compile and execute programs that is not dependent on any platform related restriction or complication.

Currently there are five known ways of providing cloud computing viz. Public, Private, Community, Combined, and Hybrid Cloud Computing. We would be implementing a cloud on which the “Smart Cloud Com” would be hosted. The “Smart Cloud Com” would be provided to the end user using a SAAS cloud. The software would contain a system that has a text editor and a terminal. The user would use the C# language to compile the program. The “Smart Cloud Com” will compile the program and return the output to the user. Additional functionalities such as monitoring of the system, user usage, user forums, and collaborative development can be added as needed.

The remainder of this paper is organized as follows: While Section 2 describes related work and problem analysis, Section 3 describes basic concept of Code Demand Compiler which is followed by section 4 i.e. System Design. Section 5 briefly describes about system architecture, whereas Section 6 is about the final implementation and conclusion is explained in Section 7.

II. RELATED WORK AND PROBLEM ANALYSIS

Given below is a brief review on recent researches which are done on compiler and cloud computing.

2.1 Problem Analysis

Compilers are used to run programs and convert them from a text format to executable format. A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters. Also once a program is compiled, it becomes platform dependent. It is also not easy to carry the same program code to multiple systems if situation doesn't permit the usage of a single system. Another drawback is that, we would need to install different

compiler on each language on which we wish to work.

2.2 Related Work

Aamir Nizam Ansari, Siddharth Patil, Arundhati Navada, Aditya Peshave, Venkatesh Borole -
“Online C/C++ Compiler using Cloud Computing” [1]: Cloud computing implies a service oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership and on demand services among other advantages. The National Institute of Standards and Technology (NIST) defines “Cloud Computing as a model for enabling easy, on - demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Some of them are lower costs, better computing, location independence, better security (although this advantage is clouded with doubts of loss of some sensitive data).

Shuai Zhang Shufen Zhang Xuebin Chen Xiuzhen Huo, —“Cloud Computing Research and development Trend” [2]:

The concept of computing comes from grid, public computing and SaaS. It is a new method that shares basic framework. The basic principles of cloud computing is to make the computing be assigned in a great number of distributed computers, rather than local computer or remote server. This article also introduces the application field the merit of cloud computing, such as, it does not need user's high level equipment, so it reduces the user's cost. It provides secure and dependable data storage center, so user needn't do the awful things such as storing data and killing virus, this kind of task can be done by professionals. Users can enjoy the service even when he knows nothing about the technology of cloud computing and the professional knowledge in this field and the power to control it.

Chunye Gong Jie Liu Qiang Zhang Haitao

Chen Zhenghu Gong,—"The Characteristics of Cloud Computing"[3]: The characteristics of cloud computing are much more complex. There are nineteen characteristics which can be used to distinguish cluster, grid and cloud computing systems. Cluster's resources are located in single administrative domain with single entity. Resources of grid system are distributed and located in administrative domain with multi entity and management policies. A cloud computing platform possesses characteristics of both cluster and grid. The cloud computing platform provides services to users without knowing much about the infrastructure. These service oriented, loose coupling, strong fault tolerance, business model and easy use are main characteristics of cloud computing. Clear insights into cloud computing will help the development and adoption of this evolving technology both for academia and industry. In practice, there are many cloud computing systems with their own characteristics. Amazon EC2 etc. supply their infrastructure as a service. Google App Engine and Microsoft supply their platform as services. In academia, there are many cloud computing projects under constructing or fully run.

Shufen Zhang Shuai Zhang Xuebin Chen Shuangzhuo—"Analysis and Research of Cloud Computing System Instance"[4]: Cloud computing can be viewed from two different aspects. One is about the cloud infrastructure which is the building block for the upper layer cloud application. The other is of course the cloud application. By means of three technical methods, cloud computing

III. CODEDOM AND COMPILER PRELIMINARY

3.1 Codedom
C# is the new flagship language in the Microsoft .NET platform. C# is an attractive vehicle for language design research not only because it shares many characteristics with Java, the current language of choice for such research, but also because it's likely to see wide use. C# is a .NET language. Therefore, distributing any C# executable to other machines requires the .NET framework to be installed on each execution machine. There are a variety of options available for distributing the .NET framework. C# and Visual Basic .NET are the two primary languages used to program on the .NET framework. The .NET framework includes a set of classes in the CodeDOM namespace, called the Code Document Object Model. T

These classes make it possible to generate source code for .NET languages without knowing the target language beforehand. There are several providers included with .NET: CSharp, VB.NET and JavaScript. Oxygene comes with its own CodeDom provider. Another feature of the provider is to compile the input, given in the document model, to a compiled assembly either directly to memory as a dynamically loaded and generated assembly, or as an executable. ASP.NET uses this mechanism to compile the ASPX pages on demand when the user first requests them. CodeDom is limited in the constructs that are supported, extended language features like for each loops or language specific options are not allowed. CodeDom also features a Code Parser; this parser is used by the designer to read back the generated code to a form. The Code Parser class is fairly limited, in that, it can only support the features exposed by the CodeDom namespace, any other constructs are unsupported. CodeDom stands for the Code Document Object Model. CodeDom allows for representing source code in an abstract data structure. Such a representation is called a CodeDom tree as the underlying data structure uses a tree paradigm. This abstract representation can then be generated into different programming languages. Each language vendor provides its implementation of the CodeDom interfaces; this implementation is called a CodeDom Provider. Programmers can then build tools that can work with many different languages without having to know about each one, instead they just delegate code generation to each CodeDom Provider. The CodeDom interfaces also expose types and methods that allow for parsing source code and creating the corresponding CodeDom. The current implementation of the Eiffel CodeDom Provider does not support parsing. Code generation is a much more popular usage of the CodeDom.

3.2 Compilers
A compiler is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language which is the target language, often having a binary form known as object code. The compiler takes source code files that are written in a high-level language, such as C, BASIC, Java or C# and compiles the code into a low-level language, such as machine code or assembly code. This code is created for a specific processor type. The program can then be recognized by the processor and run from the operating system. After a compiler compiles source code files into

aprogram, the program cannot be modified. Therefore, any changes must be made in the source code and the program must be recompiled. Fortunately, most modern compilers can detect the changes to be made and then need to recompile the modified files, which in turn saves the programmers a lot of time.

The most common reason for wanting to transform source code is to create an executable program. The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instructions. An important part of any compiler is the detection and reporting of errors; Compile-time errors are special. With a runtime error, your program may be causing trouble in the world. But with a compile-time error, the problem never progresses to that point. These errors improve program quality.

IV. SYSTEM DESIGN

Compiler is an online editor that lets the user to write the code all from the comfort of the user's browser. Compiler is much more than just an online compiler, it's a cloud infrastructure that compiles and executes the user's code allowing the user to interact with the application in real-time. Using compiler as service over cloud, all programs are stored at server side and also compiled at server side. The user can login and then write the code in window or directly import from another location of his local drive. Then by using the "smart Cloud Com" the user can compile C# code. If there is any error it will be generated and the report will be displayed on client window

CodeDom (Code Document Object Model) Technology is applied to generate Centralized C# compiler using Cloud Computing in 3 tier architecture.

□ Data Layer (Back End): Available in the Web Server which contains account information about the user.

□ Business Layer (Middle End): Decision making

layer from the application layer.

□ Application Layer (Front End): It acts as a User Interface, showing data to the user, getting input from the user.

V. SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more view of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships between them.

The system uses a dual-layered architecture. The lower layer consists of clients, which are of lower configuration. The upper layer consists of the server the primary functions of four project are:

□ Registration: - It accepts the details of a new user and stores it in the User Details database. This action is logged in the Logs database which in turn ensures that the user is registered before the first login.

□ Login Option: Using this option, the user can login into the cloud of the Cloud Vendor. Then the user is provided with login id and password. By using these details, he can login to that cloud from any network.

□ Compile option: This would take the code in the text box to the server side for its compilation, and at the server side the compiler package has been imported.

□ Execute Option: The user code is being executed and result will be displayed on client window.

The System Architecture of the proposed system is as shown below:

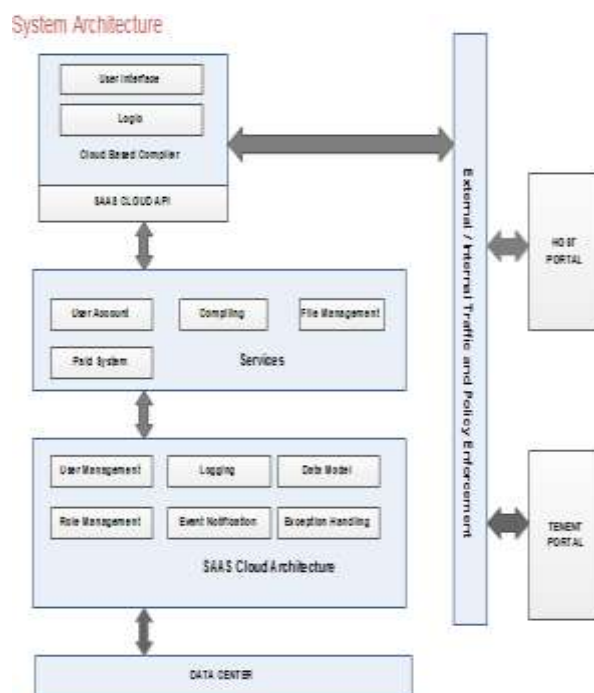


Fig2: System architecture

VI. FINAL IMPLEMENTATION

This project uses CodeDOM technology to generate Centralized C# Compiler. The CodeDom provides a language independent object model for representing the structure of source code in memory. The System.CodeDomNamespace contains number of classes, interfaces and enumerations. The user(s) paste/write the code in the main window and compile it and execute the code. By using smartcloud.com we can conduct online practical examination too.

The .NET Framework provides a feature called Code Document Object Model (CodeDOM) that enables the output of source code in multiple programming languages at runtime, based on a single model that represents the code to render. We can generate assemblies dynamically at runtime and execute. ASP.NET uses the CodeDOM to create object graphs that it compiles into assemblies that can render HTML pages and controls. It was assumed that the user will use his or her favourite text editor to create and correct program files. This assumption allows to create a very simple front-end that loads quickly and is platform independent. Although the front end is designed to be as simple as possible with only a few commonly used options, it is sufficiently functional and can be used quickly.

The System.CodeDom.Compiler namespace contains enumerations, interfaces, and classes used to generate and compile source code. It also checks whether the text are empty or not. It uses CompileResults class to represent the result of compilation that are returned from a compiler, CompilerError class to represent a compiler error or warning and CompilerParameters class to represent the parameter to invoke the compiler. After successful compilation, the compiler will generate either dll or exe file. This exe file produces the desired output for the given source code.

The compilers are hosted on virtual machines created in cloud. The path is set to these compilers using environment variables on the virtual machine. The CodeDOM provides types that represent many common types of source code elements. A program can be designed which in turn builds a source code model using CodeDOM elements to assemble an object graph. This object graph can be rendered as source code using a CodeDOM code generator for a supported programming language. The CodeDOM can also be used to compile source code into binary assembly. Some common uses for the CodeDOM include:

Templated code generation: generating code for ASP.NET, XML Web services client proxies, code wizards, designers, or other code-

emitting mechanisms.

Dynamic compilation: supporting code compilation in single or multiple languages.

System.CodeDom Namespace The System.CodeDom namespace contains classes that can be used to represent the elements and structure of a source code document. The classes in this namespace can be used to model the structure of a source code document that can be output as source code in a supported language using the functionality provided

by the System.CodeDom.Compiler namespace.

System.CodeDom.Compiler Namespace: The System.CodeDom.Compiler namespace contains types for managing the generation and compilation of source code in supported programming languages. Code generators can each produce source code in a particular programming language based on the structure of CodeDOM source code models consisting of elements provided by the System.CodeDom namespace.



Fig3:Homepage of Smart Cloud Com

Smart Cloud com is a cloud based compiler in which user can use this cloud based compiler using <http://coc.myprojectdemo.in/>. The above is the home page in which user can login. A new user has to register and then, can login.



Fig4:LoginPage



Fig5:Code Window of the Centralized C# Compiler

Once the user login, he obtains the above window in which the user has to give the code name and choose the file or can directly write the code. Then the user has to click on the compiler button. Then it will register in the back end and at the same time, the code will run in the windows form

as shown in the fig6. There it will get executed.

If there is any error, the error will be displayed in the user's system else exe file will be generated. Later the user has to click on the exe file to get the result.

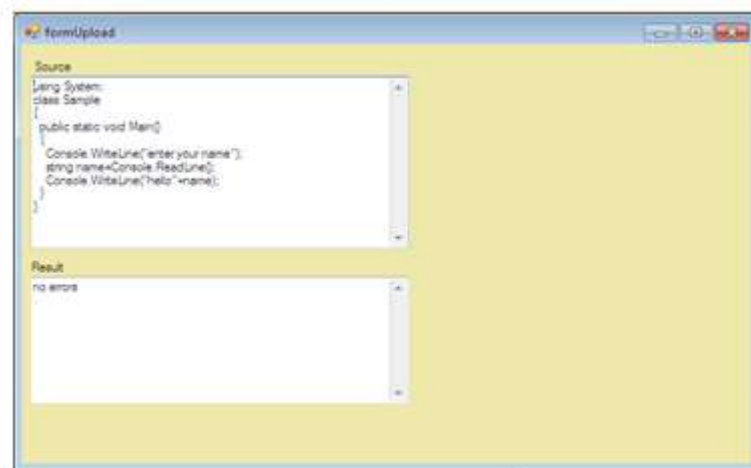


Fig6: Window form of the code (No error in the Source Code)

Compile option would take the code in the text box to the server side for its compilation and at the server side, the compiler package has been imported. Finally the output of the program is displayed as shown in fig7.



Fig7: Output Window

VII. CONCLUSION

A compiler, which is the heart of any computing system, transforms source code from a higher level language to a lower, machine level language. This is mainly done in order to create

executable files which can then be run in order to execute the program and its instructions. As compared to the current scenario, where each compiler is required to be installed on each machine separately, this would

eliminate the need to install compilers separately. So, we can check our code at the centralized server. Another advantage of such a project is that whenever the compiler package is to be upgraded, it can be done easily without again installing it on each and every machine. The main reason for creating the project is to provide a centralized compiling scheme. Also, it will act as a centralized repository for all the codes written. The other major advantage that this system will have over the others is that it will make the users' system lightweight i.e. there will be no need to maintain separate compilers at the client's side. Also, the process of maintenance and distribution of dynamic usernames and passwords will be greatly simplified. Also, authentication and personalized task distribution will be made possible.

us/develop/net/tutorials/getstarted.

REFERENCES

- [1]. Aamir Nizam Ansari, Siddharth Patil, Arundhat i Navada, Aditya Peshave, Venkatesh Borole , Online C/C++ Compiler using Cloud Computing, Multimedia Technology (ICMT), July 2011 International Conference, pp.3591-3594.
- [2]. Shuai Zhang, Shufen Zhang, Xuebin Chen, Xiuzhen Huo, —Cloud Computing Research and Development Trend, Future Networks, 2010. ICFN'10. Second International Conferences.
- [3]. Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, Zhenghu Gong, —The Characteristics of Cloud Computing, Parallel Processing Workshops (ICPPW), 2010, 39th International Conference.
- [4]. Shufen Zhang, Shuai Zhang, Xuebin Chen, Shangzhuo , —Analysis and Research of Cloud Computing System Instance, Future Networks, 2010. ICFN'10. Second International Conference.
- [5]. "Challenges in deploying SaaS applications", Imaginea Inc. whitepaper.
- [6]. "Future of cloud computing", www.roseindia.net.
- [7]. Donovan Kretsman, "SaaS | don't let the Cloud rain on your parade", www.focalscope.com's blog on SaaS.
- [8]. "Advantages of SaaS", www.cloudtweaks.com.
- [9]. M. Tim Jones, "Anatomy of a cloud storage infrastructure", www.ibm.com.
- [10]. "Quickly build and deploy Software as a Service applications", Ironspeed. Inc. whitepaper.
- [11]. "Tutorials Windows Azure" <http://www.windowsazure.com/en>