

C/C++ Cloud Compiler Using MainFrame

Sagar B. Tambe, Prof. Shiv Sutar, Mr. Mahesh D. Nirmal

Abstract: - This paper reports the successful use of Web-based interface to C++ compilers using mainframe. There are several benefits that make networked software desirable. A web-based application can be used remotely throughout any network connection. Any operating system can be used to access it, making it platform independent. There is no local installation or maintenance work necessary. Access can be controlled and limited if required by software license agreement. Internet Compilers Package (ICP) is an example of a networked software-engineering tool. Different cloud compiler packages can be used for various stages of application development providing either maximum error detection or code optimization using mainframe.

Keywords: - Could Compiler, Mainframe, Centralized

I. INTRODUCTION

The computer system requirements thus allowing for their use during the process of software development frequently more than one compiler package is required. Some products are known to be very useful for locating errors or debugging, while others perform extremely well when a program or library is in the final stage of development and should be optimized as much as possible. Also when facing obscure error messages, which may result in a time-consuming search for the error, a different error message from the second compiler frequently cuts that time dramatically. Therefore students should be to some extent exposed to different compilers at some point in their software courses curriculum. Although all necessary software is installed in the computer laboratories, most students prefer to work on their computers at home or dormitory and connect to the university network. That situation creates unnecessary burden either for the network administrators who have to install additional software on many machines of non-standard configuration, or on students who must purchase and install on their own several software packages along their full course of study. In order to solve the problem at least partially in the area of programming a software package was developed that allows for Web-based interfacing of various compilers. Three compilers: GNU, Microsoft, and Borland (Inprise) are used in the Bradley University intranet. Web-page based front end allows to access on different operating system platforms and also on older machines with lesser performance. Access to selected commercial software components is enabled based on the user's computer name and IP address, and also limited by password system.

INTRANET ARCHITECTURE

During software development it is important to justify which part of the software should run on the client

machine and which part should run on the server. Client side programs - applets are transferred through network when requested and execution is performed entirely on the client machine that made the request. This allows for sharing the computational cost between the server and client. It approach can be used when programs to be transferred to users are moderate in size or are be cached on client machine, or the data to be transferred between server and client, in case the application is run on the server, are very large in volume. In case of platform independent solutions, such as Java, causing lesser computational performance may be prohibitive With CGI much less information has to be passed to the server. The server executes instructions based on the given information and sends the results back to the local machine that made the request. This is used in the opposite cases, when the software package is large or should not be released to user, or when amount of data to be transferred is small. However, large number of clients that access the server simultaneously would make CGI-based approach undesired

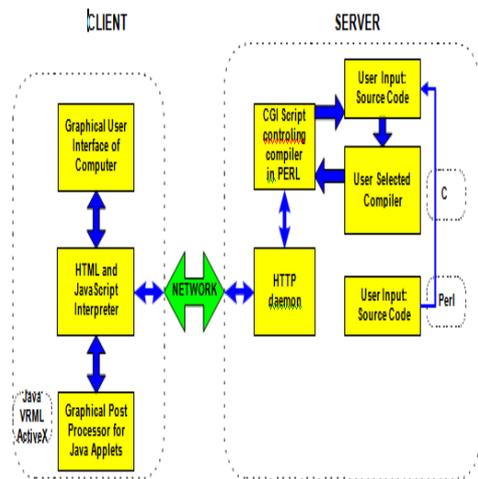


Fig1.2Data flow in an Internet Compilers using mainframe

These software design problems were considered and solved in the ICP. The user interface is programmed in HTML enhanced with JavaScript. The purpose of the project was allowing students to get familiar with different compilers and compiler optimization techniques rather than make another huge GUI application to wrap compilers. Therefore, it is assumed that the user will use his or her favorite text editor to create and correct program files. This assumption allowed to create the very simplified front-end that loads quickly and is really platform independent that is shown later in Fig. 2.

The server side part of the application is implemented using CGI scripts written in PERL that handle communication between a user and different compilers. The data flow of the complete application is shown in Fig. 1. That script does the file managing, runs compilers and processes the compilation results. The result is both the source code listing and a binary code to download or a list of errors sent back to the user. To use ICP, paste the program code from your compiler text editor, or from any text editor, to the web page form. Then submit the form. The compilation will be performed by PERL script on the server in batch mode. 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 PERL script located on the server has to deal with the translation of these common options to the actual options of compilers from different vendors. It also handles the compilation errors and processes the report to the server side for its compilation and at the server side the compiler package.

II. ONLINE COMPILER USING MAINFRAME

The primary functions of our project

2.1 Compile option: This would take the code in the text box will only need to do the steps in this section once except as otherwise noted Even if you will be installing multiple

1. Define a load library and upload the cloud compiler software. Preparing for Installation and Installation Cloud Compilers .

2.2 Execute Option: The user is provided with the links of all the executable files that were present in his or her folder and were already compiled at least once without errors.

2.3 Start test option: Till this button is not clicked the test does not start and the student cannot start writing the code. All programs and their timestamps of when were they compiled are stored at the server side database. OCC is an online compiler cum interpreter, and a simple collaboration tool. It is a pastebin that executes code for the user(s). The user(s) paste/write the code in the main window and the OCC compiles it and stores an executable file in its database. This .exe file can then be accessed directly and downloaded to the user's terminal using a URL provided by the OCC itself. The feature of downloading the executable file onto the user's terminal ensures that malicious codes (for example: code to format the C: drive on the server itself) written on the server will not execute on server itself (thereby keeping the server intact & safe.

2.4 Dot Net Technology

According to Microsoft, the developer of MS.NET, 'the .NET framework is the Microsoft Web services strategy to connect information, people, systems, and devices through software. Integrated across the Microsoft platform, .NET technology provides the ability to quickly build, deploy, manage, and use connected, security-enhanced solutions with Web services. .NET-connected solutions enable businesses to integrate their systems more rapidly and in a more agile manner and help realize the promise of information anytime, anywhere, on any device.

III. PROJECT ARCHITECTURE

Setting Up Server Mainframe(s):-

Even if you will be installing multiple Cloud Compilers you will only need to do the steps in this section once. Define one or more RACF user IDs on the server mainframe.

2. Select a job class to use for compiles.
3. If you will be using PassTickets, perform the steps in Server System PassTicket Configuration Tasks.
4. If you need more than one set of compiler defaults for a given compiler then refer to Appendix VI, Configuring.

Setting Up the Client Mainframe(s):-

1. Repeat for each cloud compiler you are installing
2. If you will be using PassTickets for authentication then perform the steps in Client System PassTicket Configuration Tasks.
3. If using passwords (keys) for authentication then perform the steps in Configuring Keys.
4. Customize the System Parameter File. System Parameter File.

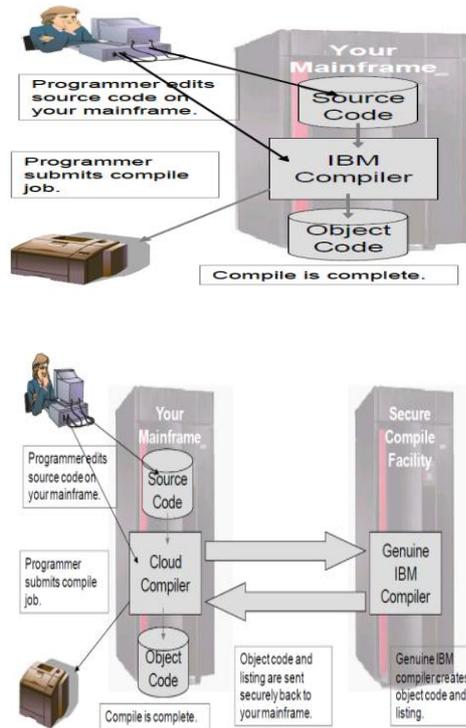


Fig 2:-Cloud Compiler Installation

IV. ADVANTAGES

This new generation of technology is based on Web services small building-block applications that can connect to each other as well as to other, larger applications over the Internet .NET combines unprecedented developer productivity with performance, reliability, and deployment. .NET makes building real world Web applications dramatically easier. .NET server controls enable an HTML-like style of declarative programming that let a person build great pages with far less code. Displaying data, validating user input, and uploading files are all amazingly easy. Some of the other main advantages are: Interoperability, Common Language Runtime Engine, Language Independence, Base Class Library, Simplified Deployment, Security and Portability.

V. DISADVANTAGE

Applications using the MS .NET framework tend to use much of the system resources and especially those which are running on Microsoft framework. Also, loss of trade secrets and by passing of license is the major problem caused by reverse engineering. Regular garbage check and collection makes the application pause for sometime from execution.

VI. CONCLUSION

By integrating and enhancing the capabilities of these essential technologies, we hope to introduce the 'Online Compiler using mainframe' and to contribute to the current examination system. It would basically be a platform for students of the university to give their practical examinations online. There would be a cloud where there will be a server which would have the power to compile the student's code stored on another machine. As compared to the current scenario where each machine should have the C/C++ cloud compiler installed separately and an examiner has to visit each machine to check each and every student's code. This would eliminate the need to install compilers separately, the examiner does not need to visit each student but can check the codes at the centralized server as well as each student's records maintained for future references. 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.

REFERENCES

- [1] Wall, L., Christiansen, T., Schwartz, R.L. Programming Perl, O'Reilly
- [2] Sweet, W. and Geppert, L., "http:// It has changed everything, especially our engineering thinking," IEEE Spectrum, January 1997, pp. 23-37.
- [3] Camposano, R.; Deering, S.; DeMicheli, G.; Markov, L.; Mastellone, M.; Newton, A.R.; Rabaey, J.; Rowson, J.; "What's ahead for Design on the Web", IEEE Spectrum, September 1998, pp. 53-63.
- [4] Hank Shiffman, Making Sense of Java.
- [5] Hank Shiffman, Boosting Java Performance: NativeCodeJITCompilers, <http://www.disordered.org/Java-JIT.html>.
- [6] Gundavaram, S., CGI Programming on the World Wide Web. O'Reilly & Associates, Inc., 1996.
- [7] Wall, L., Christiansen, T., Schwartz, R.L. Programming Perl, O'Reilly & Associates, Inc., 1996
- [8] Shufen Zhang Shuai Zhang Xuebin Chen Shangzhuo, "Analysis and Research of Cloud Computing System Instance", Future Networks, 2010. ICFN '10. Second International Conference
- [9] Shuai Zhang Shufen Zhang Xuebin Chen Xiuzhen Huo, "Cloud Computing Research and Development Trend", Future Networks, 2010. ICFN '10. Second International Conference
- [10] Grobauer, B. Walloschek, T. Stocker, E. Under Cloud compiler Computing Vulnerabilities Security & Privacy, IEEE March-April 2011
- [11] Chunye Gong Jie Liu Qiang Zhang Haitao Chen Zhenghu Gong, "The Characteristics of Cloud Computing", Parallel Processing Workshops (ICPPW), 2010 39th International Conference