

# Web Based Programming Environment Using Network-Virtual Cloud Compilation

Sagar B. Tambe

Department of Computer Engineering,  
MITCOE, University of Pune,  
Pune, Maharashtra, India

**Abstract** - Cloud compilation is a system for enabling convenient, on demand network access to a configurable cloud computing resources that can be rapidly provisioned and released with minimal management effort. The paper aims to describe an online compilation which helps to reduce the problems of portability and storage space by making use of the concept of cloud computing. The ability to use different cloud compilers allows a different programmer to pick up the fastest tool to compile the code and remove the errors and get expected output. Moreover, a web-based application can be used remotely throughout any network connection. The errors/outputs of the code are stored in a more convenient way. Also, no need to installing the different compiler on each computer is avoided. Different cloud compiler system can be used for various stages of application development providing either maximum error detection or code optimization.

**Keywords** - *Cloud Compiler, Centralized, Multitarget Compiler, Servlet, Encryption/Decryption, Optimization.*

## 1. Introduction

During the process of software development frequently more than one compiler package is needed. Some products are very useful for locating errors or debugging, while remaining perform extremely well when a program or library is in the last stage of development and should be optimized as much as possible [1]. Also when showing error messages, which may result in a time-serving search for the error, a various error message from the compiler frequently cuts that time dramatically. Therefore students should be to some extent exposed to various compilers at some point in their software courses curriculum. Although all required software is installed in the computer laboratories, almost students to work on their computers at home or laboratories and connect to the university network or Research center [2]. That position creates

unnecessary failure either for the network administrators who have to install additional software on many machines of non-standard configuration, or on user who must purchasing Software and install on their own several packages along their full course of study .In order to find the solution the problem at least partially in the area of programming a software package was developed that allows for Web-based interfacing of various compilers. Compilers: GNU, IBM, Microsoft, and Borland (In pries) are used in the Bradley University intranet [2]. Web-page based compiler front end allows to accessing them without any restrictions regarding the computer system requirements thus allowing for their use on different operating system platforms and also on older machines with lesser performance. Access to short listed commercial software components is enabled based on the user's computer name and IP address for particular machine, Also limited by password system.

## 2. Related Work

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. User side programs applets are exchanged through network when requested and execution is performed entirely on the client machine that made the request. This required for exchange the computational cost between the server and client. It application can be used when programs to be exchanged to users are moderate in size or are be cached on user machine, or the data to be exchanged between server and user, in case the application is executes on the server, are very large in volume [3]. In case of different platform independent solutions, such as Java or more programming , causing low computational performance may be prohibitive with CGI much less information has to

be passed to the server. The server running instructions based on the available information and sends the results back to the current machine that made the request. This is used in the across cases, when the software package is high or should not be released to user, or when size of data to be transferred is small in size. However, large number of clients that access the server simultaneously

would make CGI-based approach undesired. These software design problems were considered and solved in the ICP. The user interface to different programmed in HTML enhanced with JavaScript. The aim of the project was allowing user to get familiar with different compilers and compiler optimization techniques rather than make another huge GUI application to wrap compilers.

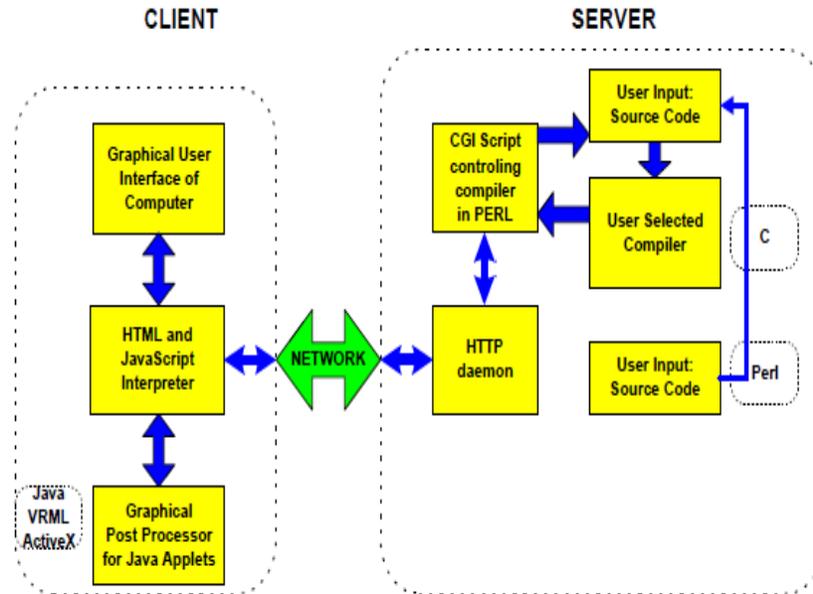


Fig. 1 Data flow in an Internet Compilers

Therefore, it is assumed that the user will use his or her favorite text editor to create and correct program log files. This allowed to create the very simplified front-end that loads quickly and is really platform independent that is shown later in Fig. 1. 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 sequentially complete application is shown in Fig. 1. That script does the file management, execution compilers and processes the compilation results. The output is twice the source code listing and a binary code to download or a list of errors sent return to the user. To use ICP, paste the program code from your different compiler textbox editor, or from any textbox editor, to the web page complicit form. Then submit the form. The compilation will be performed by PERL script on the server in batch mode for that situation. Although the front end is designed to be as simple as possible with only a few commonly used options, it is required functional and can be used quickly. The PERL script stored on the server has to deal with the

translation of these common options to the actual options of compilers from different vendors. It also maintain the compilation errors and processes the report to the server side for its compilation and at the server side the compiler package [4].

### 3. Multitarget Compiler

A program that convert between high-level languages is usually called a language converter. A different language rewriter a program that translates the form of expressions without a change of language. A different compiler is to perform many or all of the following operations: lexical analysis for compiler, preprocessing, parsing tree generation, semantic analysis (Syntax-directed translation), code generation, and code optimization for target program. Program error caused by incorrect compiler behavior can be very difficult to track down and work around therefore, compiler implementations invest significant effort to ensure the correctness of their software. A compiler is a computer instruction (or set of programs) that transforms source code written in a

programming language (the source language) into another computer language (the target language, often having a binary digit form known as object code). If the compiled program can run on a computer whose CPU or operating system is different from the one on which the compiler runs, the compiler is known as a cross-compiler [1]-[2]. And stored the data a program that exchange from a low level language to a higher level one is decompile. Two types of log are present. One is for keeping information about user login logout. It has parameters such as username, login time, and IP address. Another log is

created for storing compilation information such as filename, username, time of compilation, status of result of compilation. Database will be provided for user authentication and log maintenance purpose. Only administrator is having access to database at server side. Another open source application known as Dist cc is made as desktop application but it works only for 'C' and 'C++' not for java. It is basically a distributed compiler which distributes the task of compilation on no of servers [5].

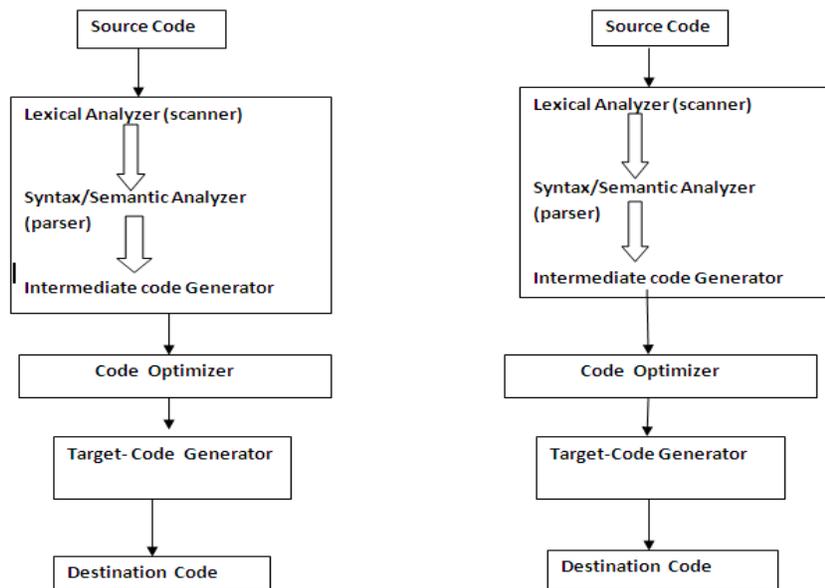


Fig. 2 Multitarget compiler.

The most common reason for wanting to transform source code is to create an executable program. The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language.

### 3.1 Source File

This is a normal text file made by Text editor (IDE, like MS Notepad, Linux Vi editor etc.) for developing source files. Encryption/decryption system: This is responsible for providing network security. It generates the cipher text for the source file to be sent to the host across the network. It involves network security algorithms such as RSA (Rivest, Shamir, and Adleman), SHA (Secure Hash Algorithm), and MD-5 (Message Digest 5) etc...

### 3.2 Network Handler

It is a lower layer application based on Transmission Control Protocol (with Internet Protocol (TCP/IP) or User Datagram Protocol (UDP) to transfer the source file (cipher text) across the network. A socket is a software endpoint that establishes bidirectional communication between a server program and one or more client programs.

The created socket associates the server program with a specific hardware port on the machine where it runs so any client program anywhere in the network with a socket associated with that same port can communicate with the server.

## 4. Online Compiler

This the basic building block of the system. It accepts the source file and produces the compiled/error file.

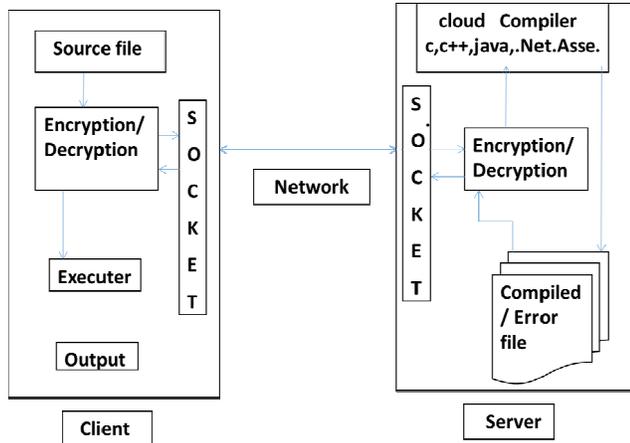


Fig. 3 online compiler

The set of files that is produced by the compiler for the respective source file. The compiled file will be having .class or .obj extension and error file as a text file [6]-[7].

### 4.1 Executer

This is the final step in the system. It accepts the compiled file(s) as input and produces the output. It is responsible for providing runtime environment for compiled files. Though some of similar applications are available in the market our application differentiating in terms of language supported, platform supported at client side and other functionality such as maintaining user directory and log files.

## 5. Centralized Cloud Compiler

To provide a centralized compiling scheme for an organization or institution. Centralized storage for all the codes written. No need to maintain separate compilers or SDK's at client side. User authentication and personalized task distribution. That is the administrator will be able to assign user-id, password and personalized tasks to all the clients.

The codes will be compiled centrally and the results will be displayed at client-side application. Both the error stream and output stream of the compiler will be captured and the output will be sent to client.

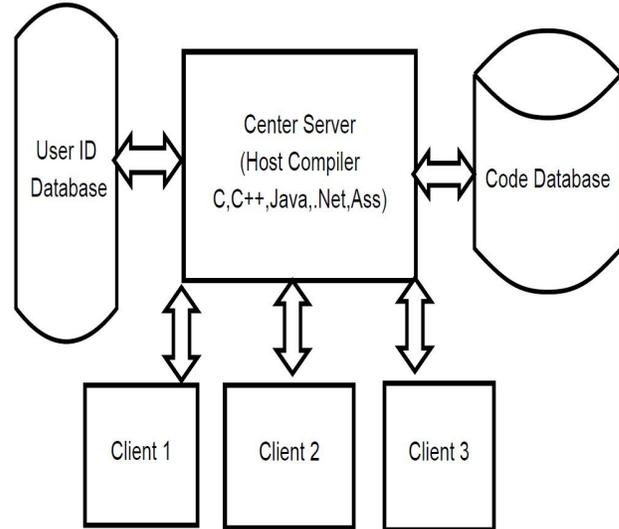


Fig. 4 Centralized Cloud Compiler

Servlets will be used for client-server communication. Tomcat Apache 5.5 will be used as server host. A database of all the codes written by the clients will be maintained. A client may retrieve the stored-code at any later instance [8]. The administrator will have full authority to compile and execute the codes stored by clients for evaluation. Statistical details of compilation time, execution time, etc. will be maintained at the server side.

Extremely efficient for educational institutions since maintenance of compilers needs to be done at only server side. Direct comparison of outputs of all the clients can be done at the server side. A client may be assigned a different task every time he logs in depending on the will of the administrator. Efficient for conducting practical examinations, since every client will be assigned a different log in id and password [9]. The administrator may create, edit and delete client profiles anytime.

## 6. System Block Diagram

User Signup/or login into the cloud through any network device and using a desktop application. On server side database serialized object is created for particular client and its store using a file. User create new code / update the existing code to on server database.

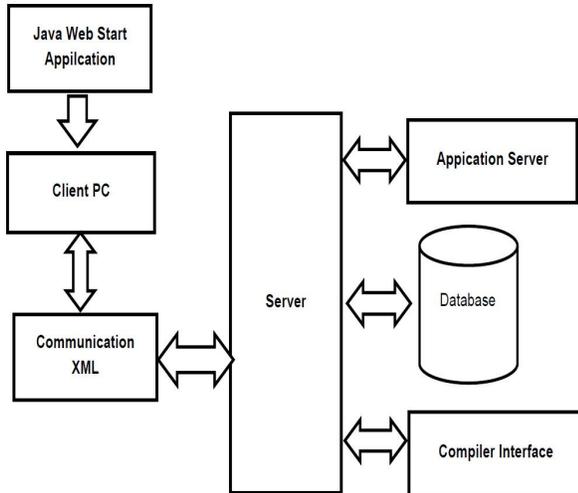


Fig. 5 System Block Diagram

On Server side particular compiler package has been imported and client code is given as input to it. Compiler compiles the client code and if any error is generated error messages are sent back to the client. If not any error message generated and program has been successfully compiled. Then user have an option to execute the code or not. If user want to execute the code, the executable file is downloaded and executed on client side. Then user can logout or continue the application.

### 7. Web Cloud Compiler:

Firstly client performs login process by submitting login Id and password. Server authenticates user. If user is first time user then server gives him user ID. According to choice selected by user, operation is performed on data or program. If user wants to encrypt the data then encryption is performed on server side and encrypted data is sent to client.

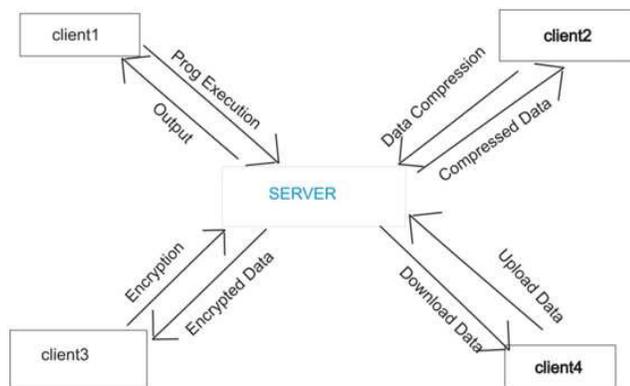


Fig. 6 Web Cloud Compiler

This paper is providing GUI to client. Firstly there will be menu displayed on the screen. Then user will select menu for which he wants to access the service. Then the data is uploaded to server and processing will be done on server side. Then output will be sent to client. Thus project is providing security for transferring the data [10]-[13].

### 8. Programmer's Design

#### 8.1 Mathematical Model

Memory utilization: The experiment was done to find space saving capability. We implemented vc for Java (jdk 1.6.0), TURBO C/C++ compilers, .NET, Assembly (TA, MA). The sizes of the remote compiler client and the server programs were 75KB and 160KB, respectively. It was required for server having Windows XP installed. Clients were working fine for Windows as well as Linux operating system. Consider a network with N systems. The software/resource to be installed is S Variable (size of S in MB). In general the total memory occupied in the network is the sum of all the memory installed on each system.

$$M_t = N * S \text{ MB.}$$

Where,  $M_t$  is Temporary variable N is Number of Systems. Consider N-1 systems to be clients and one system server (host) So N-1 system must installed with client program. So Clients occupy:  $(N-1) * S_c \text{ MB}$ . One (At least One) system must be installed with server program. So Server occupy:  $S_s \text{ MB} + S \text{ MB}$  Then the total memory occupied in the network is

$$M_{tvc} = (N-1)S_c + S_s + S \text{ MB.}$$

Then the total memory saved in this network is

$$M_{svc} = NS - ((N-1)S_c + S_s + S) \text{ MB.}$$

Percentage utilization of memory on this network using our concept:

$$\frac{((NS - ((N-1)S_c + S_s + S)) / NS) * 100}{}$$

The database size at server side will grow dynamically so it is not taken into consideration in above calculations. Number of systems in the user,  $N = 5$ ; Size of the software to be shared. SAT problem is NP Complete. The system can be reduced to SAT problem. A SAT problem takes a Boolean formula S that is in CNF in which each clause has exactly three literals. SAT is a restricted form of CNF-SAT problem.

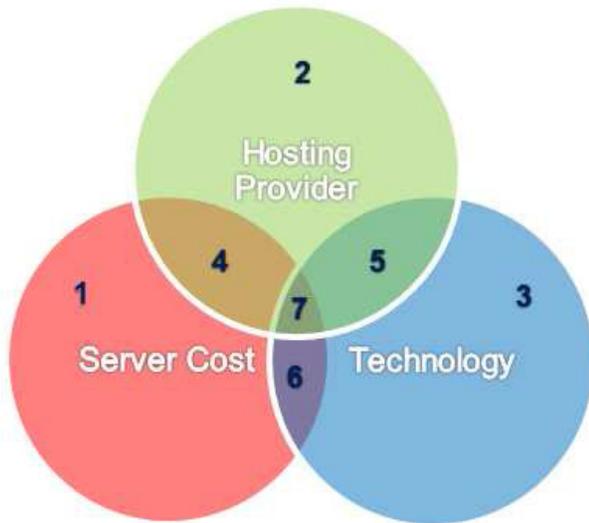


Fig. 7 Venn Diagram

### 8.2. Dynamic Programming and Serialization

A Web service is a method of communication between two electronic devices over a network. Web Services where to solve main problems, that is Firewall Traversal tree, Complexity, costing and Interoperability [11]. The W3C defines a “Web service” as a software system designed to support interoperable machine-to-machine interaction over a network. It has a described in an each machine- process able format (specifically Web Services Description Language WSDL). Remaining systems interact with the Web service in a manner prescribed by its description using SOAP related messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

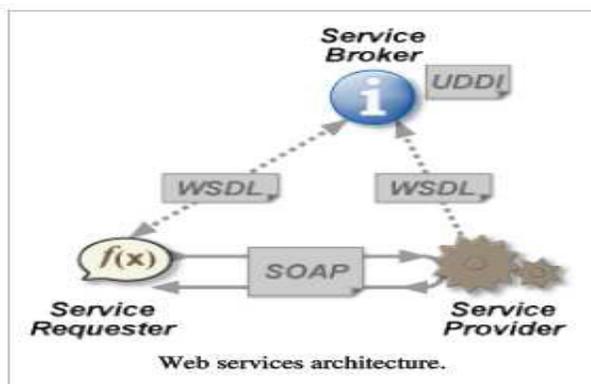


Fig. 8 Web Service Architecture

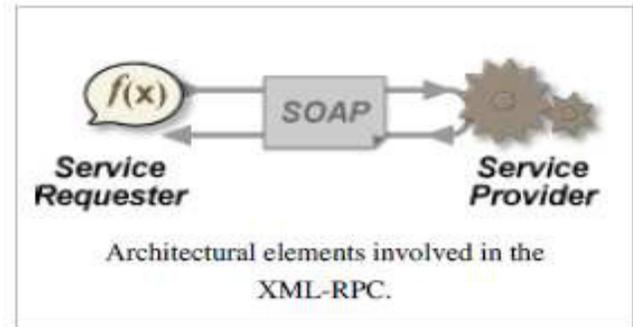


Fig. 9 XML-RPC

The W3C also states, “We can identify two major classes of Web services, REST-compliant all type Web services, in which the starting purpose of the service is to manipulate XML representations of Web resources using a uniform set of “stateless” operations; and arbitrary all Web services, in case the service may expose an arbitrary set of operations” [12]-[14]-[16].

### 9. Results and Discussion

Ensures all users are running the same current versions of all applications. Updates and patches need only be applied to the server no need to update multiple clients. No need for software to distribute software over the network.

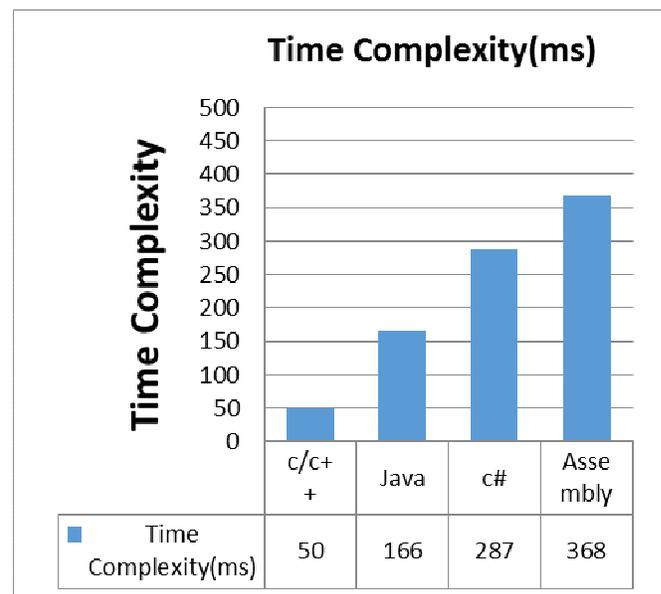


Fig. 10 Time Complexity of single compiler

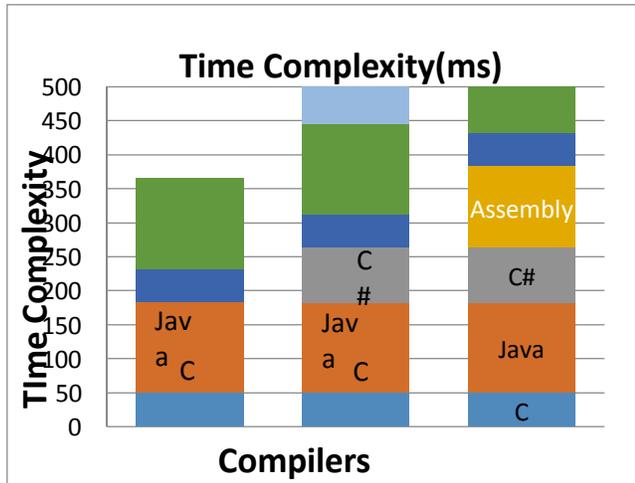


Fig. 11 Time Complexity of multiple compiler

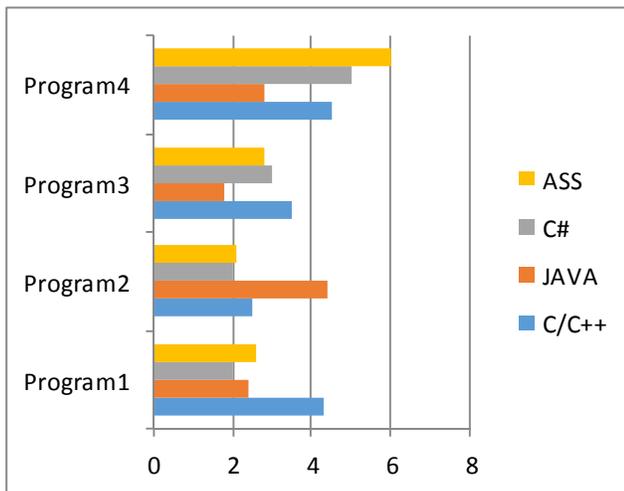


Fig. 12 Program ExecutionTime of multiple compiler

## 10. Conclusions

The main aim of this system is to develop a program which integrates various types of compiler and programming environment under one roof. The user on client machines needs to write the program in the editor. When he/she performs some operation, the code and the details of the client are sent on the network to the servlet module. At the servlet module operation is performed according to the instruction given by the client and results are sent back to the respective clients. Thus virtual compiler provides a way by which centralized control can be arranged in a controlled and reliable manner. The clients are operated in a centralized manner from the server. In future expansion point of view it can be made

for multiple clients running on different platforms. Also it can be made distributed compiler by having multiple servers handling the request. The proposed idea can be modified for developing cluster based virtual compiler for energy consumption and Accuracy [17]-[18].

## References

- [1] Aamir Nizam Ansari, Siddharth Patil, Arundhati Navada, Aditya Peshave, Pune Institute of Computer Technology, Pune IEEE, "Online C/C++ Compiler using Cloud Computing" 978-1-61284-774-0/11/2011.
- [2] Richard Perry "View/Edit/Compile/Run Web-based Programming Environment" IEEE, 0-7803-8552- 7/04 2004.
- [3] Web-based C++ Compiler. Aleksander Malinowski, Bogdan M. Wilamowski Bradley University, Peoria, IL / University of Wyoming, Laramie, WY Design on the Web", IEEE Spectrum, September 1998, pp. 53-63
- [4] A Trace- based Java JIT Compiler Retrofitted from a Method- based Compiler. (Vol 11 pp 978-1-61284, 2011 IEEE)
- [5] Vectorizing C Compiler. (vol. 8 PP 0-89791-459-7 1991 IEEE)
- [6] An Efficient Online Path Profiling Framework for Java Time Compiler. (vol 8 PP 1089- 795X/03,2001 IEEE)
- [7] Software Testing Concept Used For MAS/C-Compiler. (vol 8 PP 1089-650,2000 IEEE)
- [8] A Protocol Compiler for the Automated Implementation of Distributed Application. (vol 11 PP 1082-890,1996 IEEE)
- [9] Resource Sharing, Ponnamanda C V Varma, IEEE - Potentials December 2004/January 2005
- [10] Eliote Rusty Horold, "Java Network programming," Orielly pub, 2000.
- [11] Charles Sterling, "Microsoft Developer Network," Ink.Remote Login using TELNET, June99.
- [12] Rober Coleridge, "Microsoft Developer Network," Ink.Tier architecture. Sept 99.
- [13] Sagar B. Tambe, Prof. Shiv Sutar, Mr. Mahesh D. Nirmal," C/C++ Cloud Compiler Using MainFrame" g (IJCTEE) Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
- [14] Online: <http://support.microsoft/support/default.asp>
- [15] Compiler Optimized Remote Method Invocation,Ronald Veldema,Michael Philippsen, Fifth IEEE International Conference on Cluster Computing (CLUSTER'03)
- [16] Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps, Goce Jakimoski and Ljupco Kocarev, IEEE Transacions on circuits ans systems: Fundamental Theory and applications volume 48, No.2, February 2001.

- [17] Sagar B. Tambe, Ravindra C. Thool, Vijaya R. Thool, "Power Consumption and Congestion Control of Rendezvous Node for Wireless Biosensor Network," Published in Springer Book Chapter, Book ID: 352823\_1\_En, Book ISBN: 978-981-10-0127-7, Chapter No: 67, (ICT4SD-2015) Conference Proceedings by Springer, 2015.
- [18] Sagar B. Tambe, Ravindra C. Thool, Vijaya R. Thool, "Cluster Based Wireless Mobile Healthcare System for Physiological Data Monitoring," Elsevier Conference on Information Security and Privacy, Nagpur, Available Online At [www.Sciencedirect.com](http://www.Sciencedirect.com), Procedia Computer Science 78(2016) 40-47.