ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

A Survey on Online Cloud based Compiler for Android Smart Phones

Ketan Kardile, Pradnya Punde, Kritika Pareek

B. E., Dept. of Computer Engineering, V.A.C.O.E., Sarola Baddi, Ahmednagar, Maharashtra, India

ABSTRACT Essentially, high-level programming languages rely on certain computer programs, called compilers, to transform source code of their programs into lower level computer languages. To use a compiler, programmers are usually required to install its executable files on their machines. Local installation of compilers is indeed limited to computers and suffers from several compatibility and portability problems. For example, compilers depend on the specifications of target computers (e.g., platform, architecture, etc.), which makes it difficult for programmers to compile programs of a certain programming language on any machine they use. Cloud computing is a technology which make it is easy to access on-demand network resources in an easy manner. Mobile cloud computing is the availability of cloud computing services in a mobile ecosystem. Smart phones enable a new, rich user experience in pervasive computing. The major problem with Smartphone is that hardware resources such as CPUs, memory and batteries are still limited. This paper introduces the use of C/C++/JAVA compilers from cloud on the mobile environment which makes it easy to compile and execute programs anytime anywhere by using their android smart phone. The main aim of this application is that there is no need to install any software/compilers in their phones. Anyone can access these C/C++/JAVA compilers remotely through network.

KEYWORDS: Brute-Force Search Algorithm, Cloud Computing, Distributed Compilers, Cloud Server, Android API, Hardware, Network

I. INTRODUCTION

Typically, programmers are required to download and install the environment of their desired programming language into their own computers. This installation is actually platform dependent, which means that all prerequisite specifications of the operating system have to be satisfied. One of the major consequences of this is that all programming tasks should be accomplished through machines that already have the environment of that programming languages installed. Another concern is related to the frequent updates of the compilers (i.e. numerous version), which requires users to keep up-to-date with the refined releases as they might contain lots of useful fixes for bugs and vulnerabilities. In addition, most of the well known programming languages are available as proprietary software, which means that they are closed for extension. Cloud computing is being gradually introduced in the field of software development and recently even hit the development of embedded systems software, for example, platforms such as mBed allow to manage and build software for dedicated embedded hardware boards in the cloud further enabling convenient collaboration of large communities via flexible libraries with defined application programming interfaces. In particular, using this platform allows for new projects to be created, existing libraries and functions to be imported or edited, version and code to be shared and compiled and built binaries to locally attached prototyping hardware to be downloaded. In contrast to the standard development flow for embedded systems, however, only limited debugging facilities are available (e.g. using a “printf()-approach” to output values of variables via a suitable interface at runtime).

Open-Source Software (OSS) development, on the other hand, has acquired a considerable importance in software production nowadays. Typically, projects in this context are developed by teams of few members, or sometime by individuals. At present, several reputed software companies like Sun, Microsoft and Novell have been attracted to this trend of development. The reason behind this is that OOS can boost the usability and expendability of the products, and solve different problems of the companies, like reliability and acceleration of development and evolution. In particular, open-source compilers have attracted researchers who aim to extend programming languages or to introduce new domain-specific programming languages. Amsterdam Compiler Kit (ACK), Open64, and GNU Compiler Collection are open source infrastructures that support multiple programming languages. Polyglot and Open Java or OJ are extensible compilers for Java, while JastAdd and abc are also Java-based extensible compilers but supplied with AspectJ facilities. Being open-source, such compilers could facilitate the construction of domain-specific programming languages, extension of current compilers, and analysis of computer programs in a modular and reusable manner. Also, developers can participate in the development and improvement of such compilers. Nowadays, cloud computing environments endeavor to offer cost-efficient usage of information technology services. Cloud based services are provided on-demand based on user needs. Software-as-a-Service (also known as SaaS) is one category of the services offered by cloud computing technology. In such as service, software applications are available online so that users can run them across the network without the need for downloading and installing such applications in their machines. Users in this case need not to worry about maintenance issues since the technical support can be conducted on the cloud.

What motivated us in conducting this research is the lack of availability of such compilers in the literature that provide the combination of both: cloud and open-source computing technologies .Even though Microsoft introduced Roslyn as an open-source cloud compiler, the design and specifications of this compiler is not provided in the literature. Other issues of this compiler also induced us to come up with this work. For example, Roslyn source code is separated from its implementation. In other words, to extend Roslyn, it is required to download it source code from one of the available source-code hosting websites. On the other hand, to run/test programs on the cloud, one should go to another website to this end, without being able to access the source code of the running compiler. The contribution of this paper is concerned with a demonstration of the specifications and technical requirements needed for the design of open-source cloud compilers. Moreover, we present a detailed clarification of the prospective features and services of such kind of compilers along with their target stakeholders and the expected technical challenges.

II. Related work

The classification of the related work in this paper is divided into two main categories: Open-source compilers and cloud-based compilers. The former presents the state-of-the art compilers that were delivered in open-source distributions. The latter demonstrates the compilers that were deployed as services in cloud-based environments.

A. *Open-Source Compilers*

In the literature, different open-source compilers were introduced to increase the usability of current programming languages. The usage of these compilers has been demonstrated among researchers and developers in various purposes. In particular, open-source compilers have been utilized to extend existing programming languages, create domain-specific languages and perform source code analysis. Actually, having an open-source compiler is not a recent phenomenon. Earlier in 1983, a compiler suite called Amsterdam Compiler Kit (ACK) was proposed as one of the foremost portable compilers. Multiple programming languages are supported by ACK including C, Pascal, BASIC, etc., under different platforms. Originally, it was distributed as closed source software, but it was released in 2003 under an open source license. Based on its portability and interoperability, ACK became more usable by different kinds of programmers. We refer to for an extensive list of open-source compilers available in the market. With respect to compiler extensions and domain-specific programming languages, several open-source frameworks and infrastructures of programming language compiler were introduced in the literature. Open64 and GNU Compiler Collection are examples of open-source infrastructures for several compilers of programming language. Open64 provides compilers for Fortran77/90, C and C++, while the compilers provided by GNU Compiler Collection are for C, C++, Objective-C, FORTRAN, Java, Ada, and Go. Based on these infrastructures, various industrial and academic projects as well as domain specific compilers have been developed by diverse researchers, developers and users. Robustness, modularity and extensibility are some features that can be provided by such environments.

Concerning Java, the preferred programming language in different platforms, a number of extensible compilers have been proposed in the literature to simplify the creation of Java-based compilers. Polyglot, JastAdd , abc are good instances of extensible Java compilers. The usability of such compilers have been shown by the enormous compiler extensions accomplished based on such compilers. On the other hand, there exists a different kind of web based projects that offer open software repositories to host open-source software projects. GitHub1, SourceForge2 and

CodePlex3 are among the top and most popular alternatives of such a service. Through these websites, users can store, manage, maintain and deploy source code of their projects readily. In addition, distributed revision control, bug tracking, and shared development are examples of the features that can be provided by such services. In relevance to open source compilers, most providers of deploy their open-source compilers in these web-based hosting. It is common to have them in more than one website to increase the availability and usability.

B. *Cloud-based Compilers*

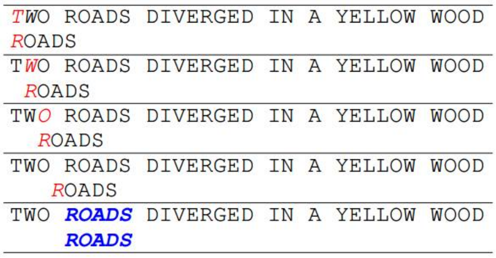
In general, cloud-based compilers provide a platform independent environment to code, compile, debug and execute software programs. This means that cloud-based compilers are free of restriction, complication and compatibility issues related to platform. Thus, users do not have to worry about downloading and installing compilers on their machines, or even keep transporting their own programs from one machine to another. Ansari proposed a cloud compiler for C/C++ programming languages. Their compiler is built as a web application using ASP.NET, which allow accessing the compiler remotely. Two main points were addressed in this compiler. First of all, centralizing the compilation process as well as the source code repository for academic and industrial organizations. Secondly, deploying the compiler in a cloud environment would simplify the maintenance and the distribution of tasks between project team members along with better authentications and simple user accounts management. The limitations of this compiler are the reduced efficiency derived by .NET framework, which consumes too many resources at the server. Also, since it is built using ASP.NET, it is a closed source compiler, which means that users cannot participate in improving the services provided by such a compiler. Another work was constructed by Raut who introduced a browser-based Integrated Development Environment (IDE) that facilitates programming, compiling, debugging, running, and testing Java programs online. It allows creating projects, sharing them with other users, and developing 2015 International Conference on Open Source Software Computing (OSSCOM) in a real-time collaboration (we refer to [19] for more information about collaborative writing on the cloud). However, this compiler is only limited to basic Java capabilities; i.e., advanced technologies like GUI, Servlets and networking are not available yet in the current implementation of the compiler. Although previous compilers are implemented using cloud computing, they are not publicly available (during the period of writing this paper) online. Therefore, we cannot conduct a fair evaluation of the services provided by each compiler.

On the other hand, in spite of the existence of other similar efforts in the literature attempting to build cloud-based compilers, they have been omitted from this study due to the reduced quality and details of their work. Numerous online compilers are publicly available through websites in the World Wide Web to compile and execute programs of diverse programming languages. Examples of this kind of compilers: CompileOnline4 Ideone5 CodeChef6 and so many others. Most of these websites allow users to create accounts, save their source-code files, compile and run them on the fly. However, relying on our testing trials of such compilers, we have observed that they only support running command-line console applications. In other words, a program that contains implementations of Graphical User Interface (GUI) components cannot be executed in such websites, even though they can be compiled in some websites. After a successful compilation of such programs, websites show runtime errors. Recently, Microsoft released its open-source cloud-based compiler called Roslyn . The main purpose of Roslyn is to provide an open-source SDK for .NET programming languages; particularly, C#, F# and Visual Basic. The ultimate goal of Roslyn is to offer the compiler-as-a-service(CAAS) model, allowing researchers and developers to customize the Visual Studio or build their own tools through the APIs supplied. Code and flow analysis as well as refactoring are examples of the APIs equipped with Roslyn. This compiler can be accessed and used through .Net Fiddle7. However, there is no adequate information about the design and specifications of this compiler in the literature. In addition, users cannot figure out the source code of Roslyn during its running. This is because that it’s actual implementation of is isolated from its source code. Expressively, to extend Roslyn or develop a new language upon it, one should download its source code from one of the available source-code hosting websites. On the other hand, compiling/running/testing programs in Roslyn on the cloud is done through another website, which is .Net Fiddle, without realizing what is happening in the compiler background (i.e., source code is invisible). To the best of our knowledge, GUI-based applications are not supported yet in the current implementation of the compiler, since it is still under development.

III. PSEUDO CODE

- Brute-Force Search Algorithm

This compares the pattern to the text, only one character at a time, until un-matching characters are found. The algorithm can be designed to stop on either the first occurrence of the pattern, or upon reaching the end of the text. To apply brute-force search algorithm, we must implement four procedures. These procedures should take as a parameter the data P for the particular instance of the problem that is to be solved, and should do the following :



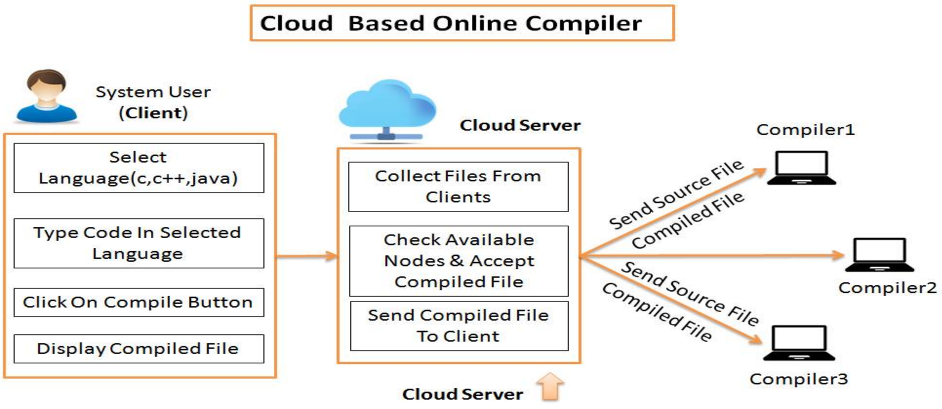
```
TWO ROADS DIVERGED IN A YELLOW WOOD
ROADS
TWO ROADS DIVERGED IN A YELLOW WOOD
 ROADS
TWO ROADS DIVERGED IN A YELLOW WOOD
  ROADS
TWO ROADS DIVERGED IN A YELLOW WOOD
   ROADS
TWO ROADS DIVERGED IN A YELLOW WOOD
    ROADS
```

- first (P): generate a first candidate solution for P.
- next (P, c): generate the next candidate for P after the current one c.
- valid (P, c): check whether candidate c is a solution for P.
- output (P, c): use the solution c of P as appropriate to the application.

IV. THE REQUIRED FRAMEWORK

Designing an appropriate framework that satisfies the needs and provides the features of open-source cloud compilers is a crucial issue. Such a framework should provide the characteristics of both: open-source computing and cloud computing.

A. *SYSTEM ARCHITECTURE*


Cloud Based Online Compiler
System User
(Client)
Select Language(c,c++,java)
Type Code In Selected Language
Click On Compile Button
Display Compiled File
Cloud Server
Collect Files From Clients
Check Available Nodes & Accept Compiled File
Send Compiled File To Client
Cloud Server
Send Source File
Compiled File
Send Source File
Compiled File
Compiler1
Compiler2
Compiler3

B. NON-FUNCTIONAL REQUIREMENTS

- Performance Requirements - High Speed System should process requested task in parallel for various action to give quick response then system must wait for process completion. Accuracy System should correctly execute process, display the result accordingly. System output should be in user required format.

- Safety Requirements - The data safety must be ensured by arranging for a secure and reliable transmission media. The source and destination information must be entered correctly to avoid any misuse or malfunctioning.

- Security Requirements- Secure access of confidential data (users details).Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction.

V. CONCLUSION AND FUTURE WORK

This system can be make more flexible if it direct access or compile the code without the internet, if the facility of our proposal inbuilt with android phones or other devices. Thus we will be implementing Code editor based on online cloud based compiler for C, C++ and JAVA etc. languages using which users can run and execute their programs anytime anywhere from their Android Smartphone.. This would eliminate the need to install compilers separately. So we can check our code at the server. Advantage of this 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. Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang,”A survey of mobile cloud computing: architecture, applications, and approaches”, Published online in Wiley Online Library (wileyonlinelibrary.com), Online ISSN: 1530-8677, DOI:10.1002/wcm.1203, 11 OCT 2011.
2. G. Roessling, H. Baer, C. Trompler, and C.M. Choi,“Mobile computing in education,” ITiCSE, 2003.
3. Abhishek Idnani, Himika Patel, Vijay Sonawane, ,“Cloud Compiler Based on Android”, Sandip Institute of Technology and Research Centre, Nasik, India
4. Mayank Patel, “Online Java Compiler Using Cloud Computing”, Mayank Patel International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-2, January, 2013
5. “Implementing SAAS: Cloud Computing and Android Based Application Framework for C Programming” IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727, Volume 11, PP 74-78, Issue 5 (May. - Jun. 2013),