RESEARCH ARTICLE OPEN ACCESS

A STUDY ON LANGUAGE COMPILER IN CLOUD COMPUTING

P.MuthuMariyappan *, K.Bharath**

**Assistant Professor, Department of CA&IT, Thiagarajar College (muthumariyappan87@gmail.com)*

*** III BCA, Department of CA&IT, Thiagarajar College (. v.n.yoooo@gmail.com)*

Abstract:

We have developed our new cloud compiler as software-as-service (SaaS), which act as a compiler that translate high level language into machine language in cloud computing . In our application, even if the client machine has no specific compilers installed, the user can write/upload a program and submit it to cloud system. Cloud system can provide different compilers for various source languages as a service, like C, C++. The program is processed in our cloud architecture and the error/output is returned back. We have proposed serial and parallel program allocation from server to backend tiers and also compared average Turnaround Time and Maximum Turnaround Time to complete all programs

Keywords — — **Cloud Compiler,compiler,machine language,Error,Turnaround Time**

I. INTRODUCTION

Cloud Computing is computing that involves a more number of computers connected through a communication networks that is similar to utility computing. Cloud computing is cost-efficient and flexible usage of IT services. The services are offered just-in-time over the internet and are paid per usage. Cloud Computing is broadly classified into three services: ”Software”, ”Platform” and ”Infrastructure” .A compiler, which is 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. A compiler translates (or compiles) a program written in a high-level programming language; which is suitable for human programmers; into a low-level machine language; which is required by computers. During this process, the compiler will also attempt to spot and report obvious programming mistakes .Using a high-level language for programming has a large impact on how fast programs can be developed.

II.RELATED WORK

Cloud compiler is defined as service oriented architecture, reduced IT services overhead for the end-user great usability, reduced cost and services. The authors proposed on online Java compiler using cloud system, the client machine doesn’t having java development kit, but still users can run Java program from his/her machine Emre Kiciman System that separates an Internet service’s logical functionality from the architectural decisions made to support performance, scalability and reliability.Cloud computing implements on decades of research in virtualization, distributed and utility computing, and more recently networking, web and software services. It 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) states that ‘Cloud Computing’ as ‘a model for enabling easy, on-demand network access to a shared pool of configurable computing resources

that can be rapidly provisioned and released with minimal management effort or service provider interaction.' It does not require the end-user to know the physical location and configuration of the system that provides these services to the end-user. The main disadvantage of cloud computing is the loss of control over the infrastructure by the users. However, this disadvantage is overcome by many an advantages that cloud computing offers. There are five known ways of providing cloud computing currently by. public, private, community, combined and hybrid cloud computing. A compiler, which is the important of any computing system, translate source greatly simplified. Also, authentication and personalized task distribution will be made possible. A Cross compiler runs on a host, but develops the machine codes for a targeted system. In embedded systems, two compilers are needed. One compiler is for the host computer which does the development, design, testing and debugging, while the second compiler i.e. a cross compiler develops the machine codes for processor of the embedded system. A compiler generates an object file for compilation of the host alone. The cross compiler switches it configuration from the host specific to the embedded system specific . Cloud computing is a technology where users connected to the network are provided on-demand services. In cloud compilers, the client writes the code in the text box provided at the interface. The code is taken at the server side for its compilation where the compiler bundle is imported.

The client gets the connections of all the executable records that are available and the code is then executed . On the analysis of these compilers we discovered:

1.Traditional C compilers are platform and mobility limited. When we write our codes on these compilers we become tied down to that machine. A code written in Windows may not result in same output as that in Linux.

2.Cross compilers configure the code written for a platform into a code specific to a targeted system.

III.LIMITATION IN EXISTING SYSTEM

1. Provide the code which having best time complexity and memory management to the contest organizers.
2. Code will be provided in different languages to the contest organizers.
3. Adopting of compiler to any website to be made possible by converting it to the web application.
4. Adopted compiler will have validity period if it expire, there will be no support provided to the website owner.
5. Adding new compiler should made possible.
6. High performance computing (HPC) should be enabled.
7. Test case to be given for each problem.

IV.OBJECTIVE OF PROPOSED SYSTEM

- providing additional functional-ties to python interpreter,
- To Generating wrapper code for c++.
- To Automating wrapper code generation for c and c++.
- Store the python interpreter in cloud service .
- Manage the turnaround time. Provide access to multiple user to access python interpreter in the cloud environment.

V.SYSTEM ARCHITECTURE:

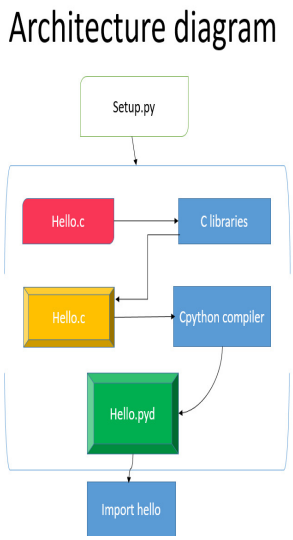

FIG 1:ARCHITECTURE OF ONLINE CLOUD COMPILER

VI.COMPILING C PROGRAM USING PYTHON INTERPRETER:

The python interpreter has the ability to run the c program.we can get c library files which are need to compile and run a c program can be done by adding cython to python interpreter. All the c library file which are need to compile a c program will be added to site packages folder in python interpreter.python interpreter cannot understand c language directly but it can understand c language if some wrapper code added to it.why it can understand is :

VI-A Nature of python vs c:

There are lots of difference between python and c .c has compiler and interpreter. Python has only interpreter. A simple example to demonstrate difference in nature of these languages. A python variable can hold different objects:

```
d = 3.2 // d holds a float
d = 'txt' //d holds a string
d = Button(frame, text='push') # instance of class Button
```

In C, C++ a variable is declared of a specific type:

```
double d;
d = 4.2;
d = "some string"; /* illegal, compiler error */
```

VI-B writing wrapper code :

Wrapper code is the code which are added to the program to run it in python interpreter.

- Every object in Python is represented by a C struct PyObject
- Wrapper code which translate between PyObject variables and plain C variables (from PyObject r1 and r2 to double, and double to PyObject).
- Once compiled it can be loaded as a Python module.

Lets look at a sample how c program can be converted into python understandable file.

```
//c program
#include <stdio.h>
#include <math.h>
#include "hw.h"
void hw1(double r1, double r2, double *s)
{
*s = sin(r1 + r2);
}
void hw2(double r1, double r2)
{
double s;
s = sin(r1 + r2);
printf("sin(%f+%f)=%f\n", r1, r2, s);
}
//c program with wrapper code
#include "Python.h"
#include "src/hw.h"
static PyObject *_wrap_hw1(PyObject *self, PyObject *args) {
PyObject *resultobj;
double r1, r2, result;
```

```
PyArg_ParseTuple(args, (char *)"dd:hw1", &r1, &r2);
hw1(r1, r2, &result);
resultobj = PyFloat_FromDouble(result);
return resultobj;
}
```

Now the above code can be executed by python interpreter. But it is impossible to write wrapper code manually to each programs.

VII.Simplified Wrapper and Interface Generation (SWIG)

Wrappers to C codes is automatically generated by SWIG. steps to create wrapper code automatically

1. First make a SWIG interface file
2. Then run SWIG to generate wrapper code
3. Finally compile and link the C code and the wrapper code

VIII C implementation for automatic wrapper Header file

```
// file: src/hw.h
void hw1(double r1, double r2, double *s);
void hw2(double r1, double r2);
```

Source file:

```
// file: src/hw.c
#include <stdio.h>
#include <math.h>
#include "hw.h"
void hw1(double r1, double r2, double *s)
{
*s = sin(r1 + r2);
}
void hw2(double r1, double r2)
{ double s;
s = sin(r1 + r2);
printf("sin(%f+%f)=%f\n", r1, r2, s);
}
```

IX. SWIG interface file

The interface file contains C preprocessor directives and special SWIG directives:

```
/* file: hw.i */
%module hw
%{
/* Everything in this block will be copied in the wrapper file. We include the C header file necessary to compile the interface
*/
#include "src/hw.h"
%}
/* list functions to be interfaced: */
void hw1(double r1, double r2, double *s);
void hw2(double r1, double r2);
```

X.Generating the module

Run SWIG (preferably in a subdirectory):

```
swig -python -Isrc hw.i
```

SWIG generates wrapper code in hw_wrap.c

Compile and link a shared library module:

```
gcc -Isrc -fPIC -I/usr/include/python3.5m -I/usr/include/x86_64-linux-gnu/python3.5m -lpython3.5m -c src/hw.c hw_wrap.c
gcc -shared -fPIC -o _hw.so hw.o hw_wrap.o
```

Note: the underscore prefix in _hw.so is required

XI build script

Make a script to automate the compile+link process

Can use pkg-config tool to extract where Python.h resides and to use the correct compiler flags.

```
# file: build.sh
swig -python -Isrc hw.i
gcc -Isrc -fPIC $(pkg-config --cflags --libs python3) -c src/hw.c hw_wrap.c
gcc -shared -fPIC -o _hw.so hw.o hw_wrap.o
python -c "import hw" # test
```

The compiled module consists of two files: hw.py (which loads), _hw.so

XIII.Testing our implementation Recall

```
void hw1(double r1, double r2, double *s)
{
*s = sin(r1 + r2);
```

}

Test:

ipython

from hw import hw1

r1 = 1; r2 = -1; s = 10

hw1(r1, r2, s)

>>> TypeError: in method 'hw1', argument 3 of type 'double *'

XIV. Specifying input/output arguments:

We need to adjust the SWIG interface file:

/* typemaps.i allows input and output pointer arguments to be specified using the names INPUT, OUTPUT, or INOUT */

%include "typemaps.i"

void hw1(double r1, double r2, double *OUTPUT);

Now the usage from Python is

s = hw1(r1, r2) 12

XV.CONCLUSION:

The cloud model described in this paper could be implemented in scenarios where a large number of users will need to compile their programs and view the output in minimal time. An example of such a scenario is online coding contests where the contestants need to submit their programs to a central server for evaluation. The number of backend compiler servers could be adjusted according to the number of users of the system. As explained increasing the number of backend compiler servers results in considerable performance improvement. By joining and enhancing the capabilities of these important technologies, we hope to introduce the new 'Online Compiler' which is 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 have the power to compile the student's code stored on another machine.

REFERENCES

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy*. San Val, 1995.

[2] Mariana Carroll, Paula Kotz':, Alta van der Merwe (2012). "Securing Virtual and Cloud Environments".

[3] In I. Ivanovet aI. Cloud Computing and Services Science,Service Science: Research and Innovations in the Service Economy. Springer Science + Business Media. AamirNizam Ansari, SiddharthPatil, ArundhatiNavada, AdityaPeshave, VenkateshBorole ,

[4] "Online C/C++ Compiler using Cloud Computing", Multimedia Technology (ICMT), July 2011 International Conference, pp. 3591-3594. Software and Information Industry Association,

[5] "Software as a Service: Strategic Backgrounder", February 2001 M Ambrust, AFox et al " Above the Clouds: A Berkeley View Of Cloud Computing", EECS Department, University Of California, Berkeley, Technical Report No.UCB/EECB-2009-28, February 10, 2009 European Network and Information Security Agency (ENISA),Cloud Computing: Benefits, Risks and Recommendations for Information Security, Nov. 2009; www.enisa.europa.eu/act/rmlfiles/deliverables/cloudcomputingrisk -assessment/at down load/full Report.

[6] Backgrounder", February 2001 M Ambrust, AFox et al " Above the Clouds: A Berkeley View Of Cloud Computing", EECS Department, University Of California, Berkeley, Technical Report No.UCB/EECB-2009-28, February 10, 2009 European Network and Information Security Agency (ENISA),Cloud Computing: Benefits, Risks and Recommendations for Information Security, Nov. 2009; www.enisa.europa.eu/act/rmlfiles/deliverables/cloudcomputingrisk -assessment at down load/full Report.