Abstract
We use compilers for executing various codes written in different languages from its text format to an executable file. These compilers have to be installed on machines using its setups depending on the language in which the code is written and also the platform on which its running. Furthermore there is large amount of wastage of space at client's side where libraries of the code have to be stacked, the code files to be executed need to be stored and also the target files where results or errors to be analysed need to be stored. Thus requiring different compilers for different languages with platform dependence. This problem can be solved by using a cloud computing model. Here we are providing SaaS private cloud for storing various compilers of different languages at server side on cloud. The client will be given option to choose a particular language compiler as per required. The code file will be sent to the compiler stored on cloud, where code will be compiled, executed and with proper handling of errors and results only the result will be displayed at client's side thus saving lot of space. In case of multiple clients requesting for code execution to the compiler on cloud, the collisions and priorities will be handled by using scheduled parallel program allocation from compiler to the back-end client. Thus taking into consideration the continuously developing technologies and languages we have developed and integrated compilers of C, C++, C#, Java, Python, Pascal etc.
1. Introduction

Cloud computing is simply management of resources and data at some third party storage space which can be accessed easily and securely from any place. Data wouldn't be required to depend on single physical machine or in that case even internal network of any industry. In older times companies had to buy licensed software’s to get any services required for the project thus increasing the estimate cost. With emergence of cloud technology companies can get large range access to services to achieve goal. Some companies even buy extra servers and databases to store the data. These expenses can be saved by using cloud technology. You don’t need to buy any high speed computer with large memory; all you have to do is get a cloud rented and sit at home using all the services of the cloud. Cloud systems reduce hardware costs on client sides. Also the difficulty of distributing software’s to individuals is avoided by using cloud. Only server side software needs to be updated.

The cloud we are using here is software as a service type (SaaS).This type of cloud allows the client to use the cloud for the period he wants. SaaS reduces costs by outsourcing the work of maintenance. The SaaS model allows the server machine to update software to be used at a single place without any need to install the same on different clients. Though there's disadvantage that you are depending totally on the security provided by some third party and the cloud service provider really see to the security part as in our project not allowing one user's code to be exposed to the other user. But with trust factor growing and professionalism this no more remains as such an issue.

![Cloud Model Diagram](cloud_model.png)

**Figure 1: Types of model of cloud**

2. Centralized Compiler

Everyone loves to get their work done in less number of penny, requiring less space and with no real latency. We have made this all possible for executing the codes of different languages parallel by providing a centralized compiler along with saving the tedious work of installing different compilers on the physical machine with complete operating system independence. With the master repository for all codes to be executed and lightweight client side system, this cloud based centralized compiler of ours will totally be on board for any developer in the industry.
3. Applying Software Engineering Approach

Many of the problems posed by the migration of computation to cloud platforms can be formulated and solved using techniques associated with Software Engineering. Much of cloud software engineering involves problems of optimisation: performance, allocation, assignment and the dynamic balancing of resources to achieve pragmatic trade-offs between many competing technical and business objectives. Software engineering is concerned with the application of computational search and optimisation to solve precisely these kinds of software engineering challenges. Interest in both cloud computing and software engineering has grown rapidly in the past five years, yet there has been little work on Software engineering as a means of addressing cloud computing challenges. Like many computationally demanding activities, Software engineering has the potential to benefit from the cloud; ‘Search based engineering in the cloud’. Cloud engineering propels us further along this journey towards a vision of software engineering in which optimisation lies at the very centre. The central justification for deploying software on a cloud platform rests upon a claim about optimisation. Cloud computing is thus the archetypal example of an optimisation-centric view of software engineering. Many of cloud computing’s problems and challenges revolve around optimisation, while most of its claimed advantages are unequivocally and unashamedly phrased in terms of optimisation objectives.

In order to best optimise cloud deployment through efficient use of available resources and minimisation of costs involved it will be necessary to predict load, behaviour, use and other profiles that affect cost and resource consumption. Fortunately, there has been much work on predictive modelling and the optimisation of predictive models using search based techniques that can be applied to cloud system behaviour prediction:

1. Scalability - There is always a need to re-engineer software, nothing goes to free. The parallelism we are providing comes into 2 types: user-level parallelism and data-level parallelism out of which we are using user-level parallelism i.e. multiple users accessing the single application.
2. Resource efficiency - The spare cycles of the past are to a great degree eliminated and therefore when and how to scale, using efficient software designs and proper use of software development life cycle becomes important.
3. Development cycles - The difficulty of distributing software to individual users is avoided by using cloud technology.
4. Risk management - First risk is that if data deployed on one cloud is to be migrated to some other cloud provider then such a migration will be quite a difficult task. Another risk that will be involved is after migration of data the old vendor may keep the data with him leading to security fullbacks.
5. Fault tolerance - Software’s should be tolerant to complete hardware failure. In our case database is provided to store the error files or the result in case of crashing of network or server for check pointing and data recovery.
6. Maintenance - Maintenance is another important field for software maintenance which is to be considered for providing services as per the agreement to the clients. The fields which are to be maintained are:
   a. Virtual machine management
   b. Managing oversubscription
   c. Translating SLAs to low-level behaviour
   d. Scalable service provision
4. Design and Implementation Constraints

Developers of the product should be aware that main feature of the intended product is portability. So they should use common libraries and tools that can work with all the common internet browser application with no problem. Developers should also be careful about the privacy of users. Since product will be cloud application, all user data will be kept on cloud server and necessary precautions should be taken to protect user data. Since product will be cloud application and all user programs will be executed on cloud server, developers should limit the privileges of the users so that they cannot harm other users’ data and system server.

5. Project Scope

Our cloud based compiler mainly aims to provide a platform to compile and execute the code that is compatible with any platform with necessary constraints, assumptions and dependencies. The compilers that we are integrating would be a compiler that is hosted on the private cloud. The use of very powerful systems will not be the need with portability provided to the developer. This project is intended to make use of ever growing technology of cloud computing in today's modern world. There are lots of IDEs available today both free source and in commercial markets. However there are lot of problems with these IDEs. Firstly it requires intensive CPU and memory usage which is not always available and due to its installation on a specific machine they lose the behaviour of portability. By using cloud computing technology, this project will remove the requirement for powerful systems and provide portability to developer.

6. System Architecture

Let us study the architecture of cloud based compiler and how the various blocks are interfaced with each other.

Cloud computing explains the concepts of distributed computing. This system makes use of the dual layered architecture in which the lower layer consists of clients, which are of lower configuration and the upper layer consists of the server. It involves Service oriented architecture.
The main aim of the project is to provide a centralized compiling scheme for institution or organization. Codes and schedule of sending codes for execution is stored on database. The Host Portal and tenant portal is interfaced via traffic environment which in turn is interconnected to Cloud Api.

The Cloud Compiler consist of Cloud Api which have a User defined interface and logic in it and other services. The other service includes compile files, file handling paid system and result display. The frontend is designed to be as simple as possible with only a few commonly used options, it is sufficiently functional and can be used quickly.

7. Conclusion and Future Scope

Thus, we have a system which will combine each above system’s advantages and will discard their disadvantages for the better future use of cloud computing. Moreover, now-a-days we require everything at ease so this all systems provide the best solution to these problems. By integrating the capabilities of these essential technologies, we hope to introduce the ‘Multi-language Compiler’ and to contribute to the current examination system. It would provide a platform for students to give practical examinations online. A cloud will be available where a server will be present which handle codes of all students and will compile codes separately sitting on another system.

As compared to the current scenario where each machine need to install compilers separately at a machine. This would eliminate the need to install compilers separately on every machine. So it will reduce memory space so that we can compile our code at the centralized server. Another advantage of the 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 multiple requests from multiple clients to execute code are also possible through this project thus increasing the execution speed and parallel execution of codes for developing a product.

References


