2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.31

An Efficient Cost Calculation Mechanism for Cloud and Non Cloud Computing Environment in Java

Monika Kherajani[1], Huma Gupta[2], Ajit Shrivastava[3]

[1]M.Tech Scholar, Truba Bhopal, India
[2,3]Asst. Prof., Truba Bhopal, India
monika.kherajani@gmail.com[1]
huma.g@trubainstitute.ac.in[2]
ajitshrivastava@rediffmail.com[3]

Abstract. Cloud-based apps can be up and running in days or weeks, and they cost less. With a cloud app, you just open a browser, log in, customize the app, and start using it. Businesses are running all kinds of apps in the cloud, like customer relationship management (CRM), HR, accounting, and much more. Some of the world's largest companies moved their applications to the cloud with salesforce.com after rigorously testing the security and reliability of our infrastructure. In this paper we proposed an efficient cost calculation mechanism for cloud and non cloud computing environment in java using SaaS, PaaS and IaaS platforms. By utilizing the aspect of cloud computing we can prove that any operational cost in the cloud environment is less in comparison to the cloud environment. In this paper we proposed an efficient approach for cloud and non cloud environment by the help of object oriented concept and their properties.

Keywords. Cloud Computing, SaaS, Paas, IaaS, Cost Computation

1. Introduction

Cloud computing is poised to revolutionize Information Technology (IT) acquisition and service models. Delivering massively scalable computing resources as a service with Internet technologies, resources are shared among a vast number of consumers allowing for a lower cost of IT ownership. Cloud computing provides on-demand computing resources dynamically, which allows companies to fundamentally change their information technology strategy.

As with any new technology, this new way of doing business brings with it new challenges, especially when considering the security and privacy of the information stored and processed within the cloud.

Cloud computing is TCP/IP based high development and integrations of computer technologies such as fast micro processor, huge memory, high-speed network and reliable system architecture. Without the standard inter-connect protocols and mature of assembling data center technologies, cloud computing would not become reality too. In October 2007, IBM and Google announced collaboration in cloud computing. The term "cloud computing" become popular from then on. Beside the web email, the Amazon Elastic Compute Cloud (EC2) [1], Google App Engine [2] and Salesforce's CRM [3] largely represent a promising conceptual foundation of cloud services. The services of cloud computing are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [4][5]. Cloud computing also is divided into five layers including clients, applications, platform, infrastructure and servers. The five layers look like more reasonable and clearer than the three categories [6].

Requirements gathering phase so far included customers, users and software engineers. Now it has to include the cloud providers as well, as they will be supplying the computing infrastructure and maintain them too. As the cloud providers only will know the size, architectural details, virtualization strategy and resource utilization % of the infrastructure, planning and design phases of SW development also have to include the cloud providers.

The cloud providers can help in answering these questions on: 1) How many developers are needed, 2) Component Reuse, 3) Cost estimation, 4) Schedule Estimation, 5) Risk Management, 6) Configuration Management, 7) Change

Management, and 8) Quality Assurance. Utility cloud computing allows users to rent Virtual Machine (VMs) from a service provider, placing an organization's sensitive data in the control of a third party. This situation places a significant level of risk on the privacy and security of the data processed by the VMs in the cloud. We propose a new management and security model for utility cloud computing called the Private Virtual Infrastructure (PVI) that shares the responsibility of security in cloud computing between the service provider and client, decreasing the risk exposure to both. In the rapidly changing computing environment with web services and cloud platform, SW development is going to be very challenging. SW development process will involve heterogeneous platforms, distributed web services, multiple enterprises geographically dispersed all over the world. In this paper we analyze several aspects of software components in cloud environment.

We provide here an overview of executing data mining services on grid. The rest of this paper is arranged as follows: Section 2 introduces problem domain; Section 3 describes about recent scenario; Section 4 shows the proposed method; Section 5 describes Conclusion.

2. Problem Domain

Cloud Architecture provides services on demand basis via internet (WWW) services. Application design in cloud computing environment or the applications which support cloud paradigm are on demand on the basis of user requirement. Those applications provide the support on various hardware, software and other resource requirement on demand.

API used in the cloud computing provide the greater advantage to provide industrial strength, where the complex reliability and scalability logic of the underlying services remains implemented and hidden in the cloud environment. Cloud Computing provide the highest utilization in terms of utilization, resource sharing, requirement gathering and utility to the other needful resources.

Traditional Computing have some drawbacks over Cloud Computing. They are:

Complicated
To work with an application, the application has to be installed on the user's computer. Then the application has to be configured for use. This is time consuming and sometimes a tedious task if the application is large. Moreover, if the application has to be updated with the latest version, it has to be done manually. So, all these things are complicated for installing each and every application to a computer.

Expensive
For example, if a company is having a website ourcompanyname.com which is hosted in some hosting server which is in United States? In the traditional way of using the servers, if a company wants more hosting server space for its website it will take time. And also if the company is not using the hosting space, it must pay for that too.

Slow
Traditional Computing is slower than cloud computing. To get a hosting space it will take some time.

Not Scalable
Traditional computing is not scalable. You cannot get as low resource as you need instantly.

Cost Scalability

Previous approach not provides cost scalability, or in other words we can say that in our approach we provide the scalability for the cost.

3. Recent Scenario

In 2009, Börje Ohlman et al. [7] analyze how cloud computing and NetInf can be combined to make cloud computing infrastructures easier to manage, and potentially enable deployment in smaller and more dynamic networking environments. NetInf should thus be understood as an enhancement to the infrastructure for cloud computing rather than a change to cloud computing technology as such.

In 2009, Vincenzo D. Cunsolo et al. [8] present the Cloud@Home paradigm, highlighting its contribution to the actual state of the art on the topic of distributed and Cloud computing. We detail the functional architecture and the core structure implementing such paradigm, demonstrating how it is really possible to build up a Cloud@Home infrastructure.

In 2010, Li Guo et al. [9] propose and present the design and implementation of Imperial College Cloud (IC Cloud). The goal of IC Cloud is to provide a generic design space where various cloud computing architectures and implementation strategies can be systematically studied. The IC Cloud design strictly follows the SOA principle and incorporates a highly flexible system design approach.

In 2010, Radha Guha et al. [10] analyses how cloud computing on the background of Web 2.0 is going to impact the software engineering process to develop quality software. As the cloud provider is an external entity or third party, how difficult will be the interaction with them? How to separate the roles of SW engineers and cloud providers? SW engineering should include framework activities to leverage all the benefits of cloud computing systematically and strategically.

4. Proposed Approach

In this dissertation we proposed an efficient approach for cloud and non cloud environment by the help of object oriented concept and their properties. For better understanding our dissertation we first present a flowchart which can clearly include all the aspects of our work. Then we present the algorithm for the proposed work and finally the explanation.

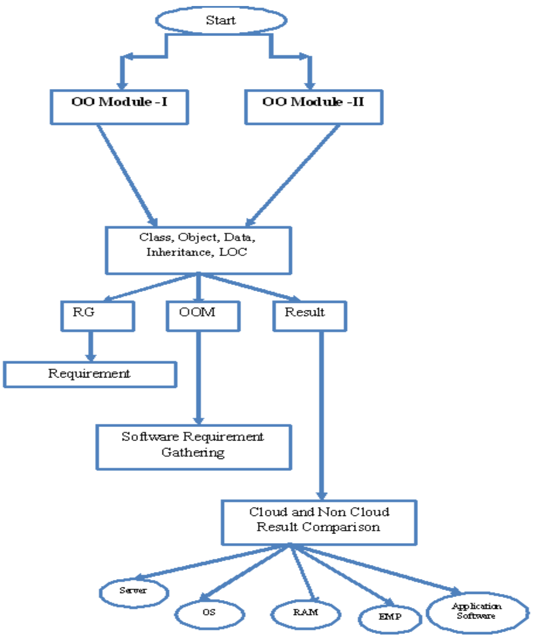

Figure1: Proposed Approach Flow Chart

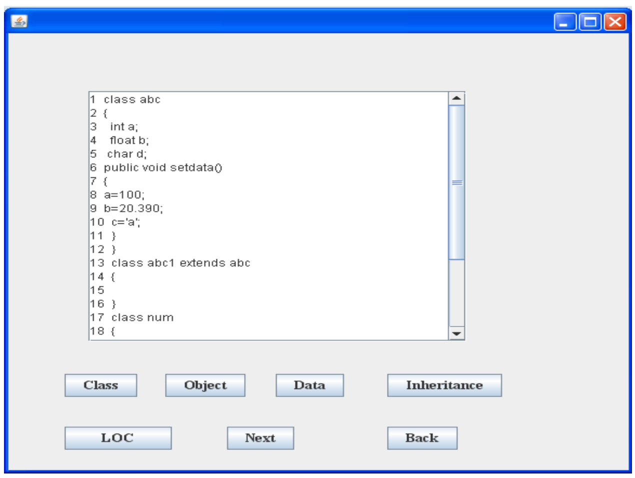

Figure 2: Initial Phase

The algorithm for our proposed work is shown below:

Algorithm:

Input: Module of C++ and Java
Output: Cost Comparison of Cloud\Non Cloud
1. Insert the module of C++ and Java
2. Calculate the OOP Parameters.
2.1 class, data, object, inheritance and a software parameter LOC
4. Requirement gathering phase (RG).
5. Object oriented module phase

For Cloud:
BOC= no. Of classes * no. Of Server * Rent per month
BOO= no. Of object * no. Of Server
BOD=no of data type * size
Total(C) =BOC+BOO + BOD

For Non Cloud:
BOC= no. Of classes * no. Of Server * actual Cost
BOO= no. Of object * no. Of Server
BOD=no of data type * size
Total (NC) =BOC+BOO + BOD

6. Software Gathering Phase
MM= 2.4*(KSDI)1.05
TDEV=2.5 * (MM)-38
Productivity=LOC/MM
Average Staffing=MM/TDEV
ACT=Average Staffing + MM
AM=MM/12

OC=MM+TDEV+ Productivity+ Average Staffing +ACT+AM

7. Cloud Cost
RG(Cloud)= (No. Of Server * Hard disk (Rent per month) + OS(Rent)+ Ram(Rent)) * no. Of month

8. Non Cloud Cost

RG(Non Cloud) =[Ram Cost + Cost*(No. Of server) + Hard Disk Cost + OS Cost * Application Software] +(no of Person + Avg Salary)* month

In our approach we can calculate the cost of cloud computing taking the consideration of object oriented programming approach. We consider two object oriented module, first for c++ and second for java module. We consider five parameters as shown in Figure 2. The parameters are class, object, data, inheritance and LOC. Then we go to the requirement gathering phase (RG). In requirement gathering phase we can calculate the number of classes, number of object, number of LOC, number of data and Inheritance.

Then we go to the next step that is object oriented module phase. In this phase we can calculate the approx cost based on class object and data. Then we select the no of server required and according to that we can calculate the cost in cloud and non cloud environment. The calculations are given below:

For Cloud:
BOC= no. Of classes * no. Of Server * Rent per month
BOO= no. Of object * no. Of Server
BOD=no of data type * size

Total(C) =BOC+BOO + BOD

For Non Cloud:
BOC= no. Of classes * no. Of Server * actual Cost
BOO= no. Of object * no. Of Server
BOD=no of data type * size
Total (NC) =BOC+BOO + BOD

Then we go to the user requirement phase. In this phase user enter the requirement according to the need and requirement. For example if we enter the no of server is 2, number of OS =1, RAM=100 MB, Hardisk=100 GB and Application Software. Then we deduce the result if we cloud with 2KDSI value. The formula for calculation is given below:

$MM= 2.4*(KSDI)^{1.05}$
$TDEV=2.5 * (MM)^{-38}$
Productivity=LOC/MM
Average Staffing=MM/TDEV
ACT=Average Staffing + MM
AM=MM/12
OC=MM+TDEV+ Productivity+ Average Staffing +ACT+AM

Then we are going to the Platforms and Storage Requirement. It is calculated on the basis of below formula:

RG(Cloud)= (No. Of Server * Hard disk (Rent per month) + OS(Rent)+ Ram(Rent)) * no. Of month

By the above formula we can deduce the cost in the cloud environment.

The same performance is checked on the non cloud environment. But the parameters and the requirement is different because of the need of cloud and non cloud. We can calculate the cost in non cloud environment by the below formula:

RG(Non Cloud) =[Ram Cost + Cost*(No. Of server) + Hard Disk Cost + OS Cost * Application Software] +(no of Person + Avg Salary)* month

After the calculation in the cloud and non cloud environment we can show in cloud environment the cost is less in comparison to the traditional and non cloud environment.

5. Conclusion

When you multiply this effort across dozens or hundreds of apps, it's easy to see why the biggest companies with the best IT departments aren't getting the apps they need. Small and mid-sized businesses don't stand a chance. In today's era with cloud computing, you eliminate those headaches because you're not managing hardware and software that's the responsibility of an experienced vendor like salesforce.com. The shared infrastructure means it works like a utility: You only pay for what you need, upgrades are automatic, and scaling up or down is easy. In this paper we analyze the cost in the non cloud environment and after that we can proof that in cloud environment the cost is less than in the non cloud.

6. References

[1] Amazon, "Amazon Web Services," http://aws.amazon.com/.

[2] Google, "Google app Engine," http://code.google.com/appengine/.

[3] Salesforce, "CRM", http://www.salesforce.com/.

[4] searchcloudcomputing.com, "What is cloud computing?" 1287881, 00.html. http://searchcloudcomputing.techtarget.com/sDefinition/0,,sid201_gci

[5] L.M. Vaquero, L.R. Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition,"

ACM SIGCOMM Computer Communication Review, v.39 n.1, 2009.

[6] Wikipedia, “Cloud computing,” http://en.wikipedia.org/wiki/Cloud_computing.

[7] Börje Ohlman, Anders Eriksson,” What Networking of Information Can Do for Cloud Computing ” 2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises.

[8] Vincenzo D. Cunsolo, Salvatore Distefano, Antonio Puliafito and Marco Scarpa ,” Volunteer Computing and Desktop Cloud: the Cloud@Home Paradigm ” , 2009 Eighth IEEE International Symposium on Network Computing and Applications.

[9] Li Guo, Yike Guo and Xiangchuan Tian ,” IC Cloud: A Design Space for Composable Cloud Computing ” 2010 IEEE 3rd International Conference on Cloud Computing.

[10] Radha Guha and David Al-Dabass ,” Impact of Web 2.0 and Cloud Computing Platform on Software Engineering ” , 2010 International Symposium on Electronic System Design.