Cost Model for SaaS Services

Thesis submitted in partial fulfilment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Tript Kaur

(Roll No. 801232030)

Under the supervision of:

Ashish Aggarwal
Assistant Professor

Dr. Damandeep Kaur
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Cost Model for SaaS services*", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Ashish Aggarwal* and *Dr. Damandeep Kaur* refer other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Tript

(Tript Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Ashish Aggarwal)
Assistant Professor

Daman

(Dr. Damandeep Kaur)
Assistant Professor

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University, Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGMENT

First of all I would like to express my deep gratitude to my supervisor Mr. Ashish Aggarwal and Dr Damandeep Kaur, Assistant Professor, Computer Science and Engineering Department, who give me a chance to explore skills in the field of Cloud Computing and spends sufficient time in exploring terms used in my thesis work. They gave me suggestions so that my research was always in progressive mode.

I am thankful to Dr. Ashutosh Mishra, P.G. Coordinator, Computer Science and Engineering Department, who always cares to my problems related to my thesis work. He always considered problem and takes immediate action for student's benefits.

I am also thankful to Dr. Deepak Garg, H.O.D., Computer Science and Engineering Department, who always considered the problem and provided each and every facility related to my thesis work.

Also, I would like to thank the members of the Computer Science and Engineering Department for their encouragement and support during preparation of this thesis report.

Most importantly, I would like to thank my parents and the almighty for showing me the right direction out of blue, stay calm in oddest of times and keep moving even at times when there was no hope.

(Tript Kaur)

801232030

ABSTRACT

The SaaS model is software delivery model that allow the end user to use the software online without it being installed on their computer. User has the access to third party infrastructure, known as Cloud, to run the application which is already installed thereon by the application provider. Unlike PaaS and IaaS models, the SaaS model does not provide the capability of controlling the operating system or hardware or network parameters. The SaaS services are open for purchase, delivered to user by provider and are used by the end user. The end user is charged by the service provider for using/utilizing the services. The charges, for using the application, are calculated by service provider based on infrastructure used, service development & deployment and its usage. Since SaaS model provides different capabilities to users than IaaS and PaaS model, so a different cost model is needed to calculate the effort of the service provider. Most of the existing Cloud cost models determine usage-charges based on infrastructure parameters rather focusing on service parameters such as performance, response timeliness, and scalability and so on. In this thesis, the cost model for the SaaS services is proposed that focuses not only on infrastructure but also on service parameters. The proposed model considers the service parameter for the SaaS such as customization, configurability, portability, security, QoS parameters including performance, usability, response timeliness, scalability and reliability, interoperability, data isolation and application isolation. With the help of proposed model, the service provider is able to calculate the efforts in developing the service more accurately and provide better service package deals to the service consumer.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER- 1

INTRODUCTION

1.1 Cloud Computing

The cloud computing is the practice that enables the end users to access the shared pool of computing resources that can be dynamically provisioned. A cloud consists of large number of interconnected virtualized or physical computers presented as one or more consolidated computing resources which can be dynamically provisioned based on service level agreements between cloud consumer and service provider [1].

1.2 Cloud computing service models:

The cloud computing offer services based on three basic service models: infrastructure as a service(IaaS), platform as a service(PaaS), software as a service(SaaS).

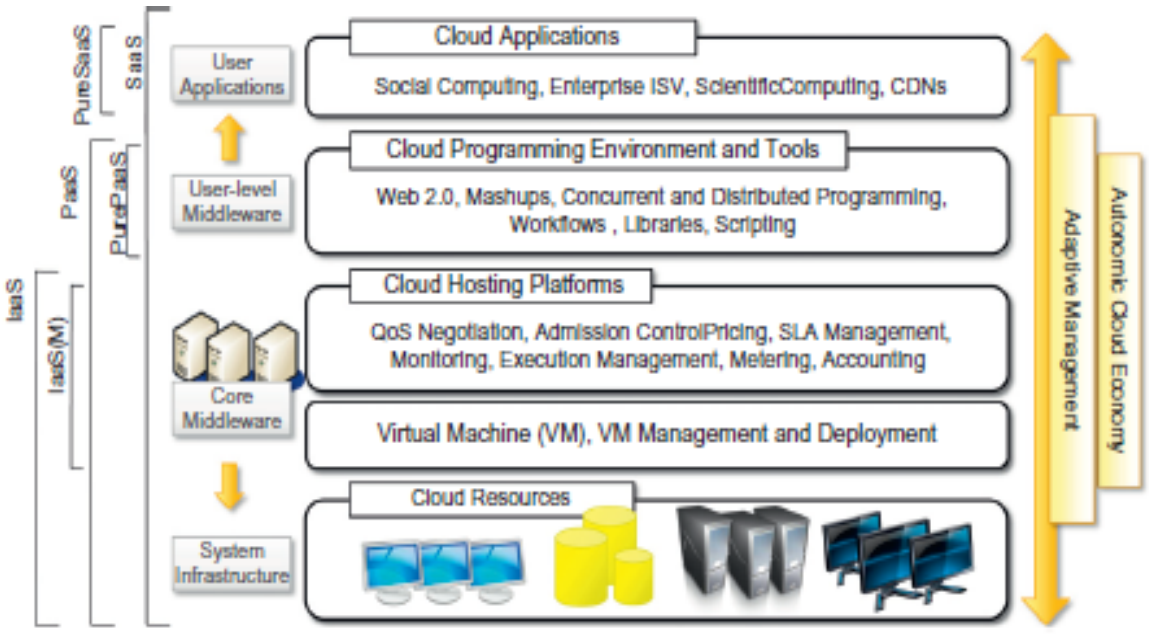

Figure 1.1 cloud computing architecture [1]

IaaS: IaaS model provide physical or virtualized hardware resources such as storage, network, processing and other basic computing resources to consumers, on the top of which they can run and deploy applications.

PaaS: PaaS model provide users a development platform for their applications. The deployment platform includes operating system, database, web server and programming language execution environment.

SaaS: SaaS model provide users the access to the software applications running on Cloud which can be accessible from anywhere at any time.

1.3 Software as a Service:

SaaS model is known for its simple deployment method, lower expenses for customer, flexible to allow many customers with single software. SaaS is accessed through web browser and charges are based on subscription of yearly or monthly. It is different from the traditional softwares in which the customers have to buy a licence for the software and to do all maintenance and installation by them. SaaS is significantly advantageous to the customers. It frees the customers from hardware, implementation, maintenance, installation costs in order to run the application on customers' side. The SaaS vendor is responsible for delivering security, performance, reliability, updates and fixes of the application.

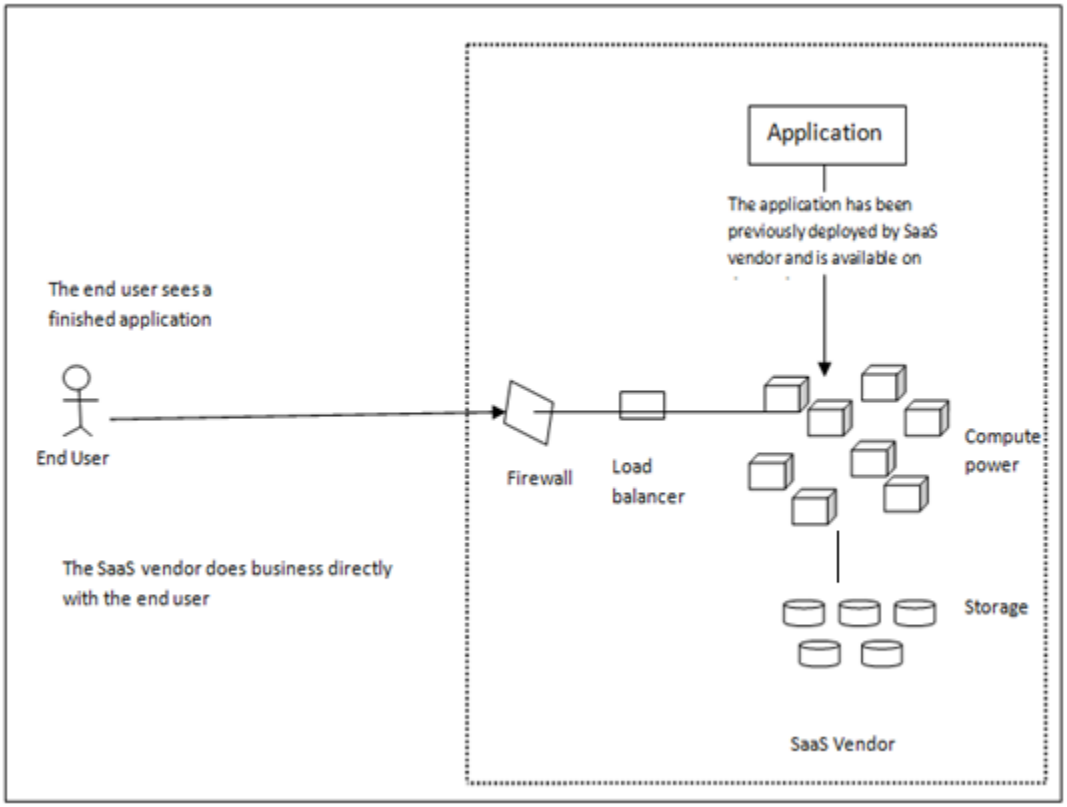

Figure 1.2 SaaS Model[28]

1.3.1Characteristics of SaaS: The following are the characteristics of SaaS [2]:

1. Security: The service provider is responsible for security of customer's data.

2. Availability: The SaaS services are accessed through web browser via internet. The customers need not to have any ownership of SaaS which is running and deployed on providers server. Through this SaaS provider achieved higher availability.

3. Customization: Customization of service allow customer to make changes to the service. The changes may be done to the functions or look or feel of the service. Some provider do not provide customizable SaaS, the user can only utilize the service.

4. Quality of service: The service provider is responsible for providing QoS features. These include performance, usability, accuracy, response timeliness and reliability.

5. Scalability: Scalability means to handle the rise or decline in load. The service provider is responsible for handling the scale up or down. The SaaS is scalable to handle enough the scale up or down.

6. Multi-tenant: Multi-tenant is a principle in software architecture in which a single software instance running on a server serves the multiple clients. In the multi tenant platform, the customers shares all components of technology stack containing servers, data model and database layers.

7. Configurability: This the key feature of SaaS. Its aspect consists user interface, data, organizational structure, workflow and business logic. The customer can opt to configure any of these.

1.3.2 SaaS architecture:

The general architecture of SaaS [3] service is given in figure. The SaaS service includes three roles service customer, platform and application provider. The service customer is end user or enterprise that consumes the service. The customer deals with SLAs which are negotiated between customers and SaaS platform. The service is delivered to customer by SaaS platform. The firmware or hardware is the lowest layer

of hardware infrastructure above which exist software kernel and virtualized management. The data tier is lowest layer of software architecture above which application tier exists. The application service in application tier provides functions such as use of configurator and configuration service by user to customize UI, data and workflow. The management layer and security/ privacy layer are important layer to SaaS. The platform data centre stores the customer's data along with other customers. Billing, metering, capacity planning, monitoring, reporting, supporting and SLA management are included in platform management.

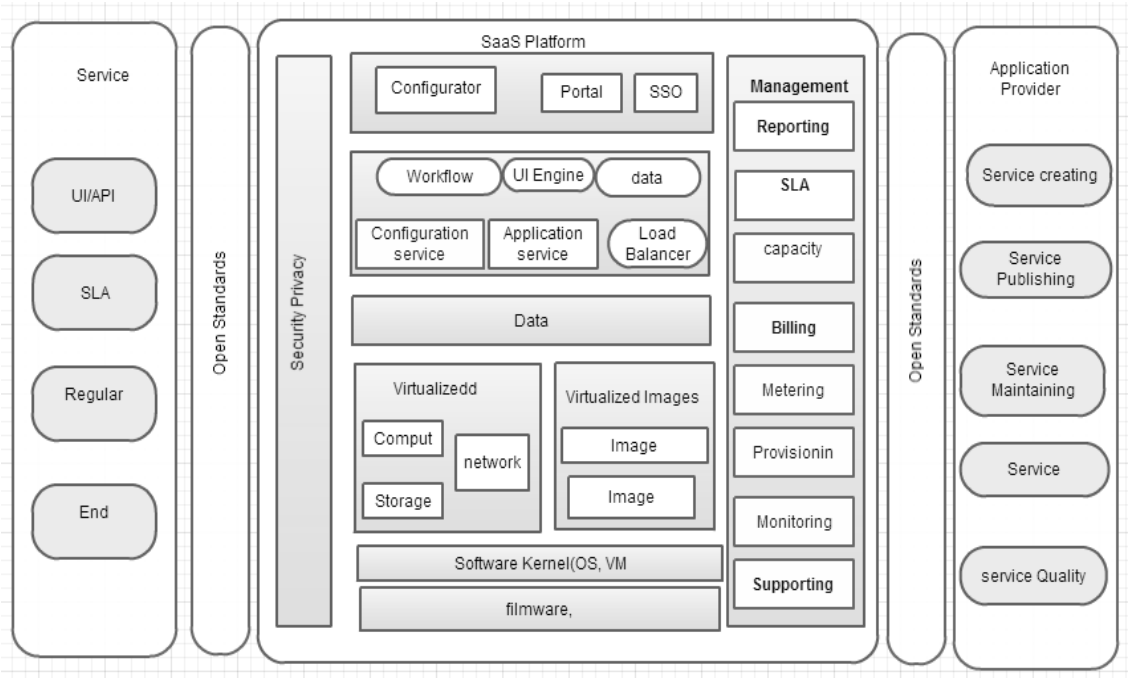

Figure 1.3 SaaS general architecture[3]

1.3.4 SaaS Pricing:

SaaS vendors generally charge their applications through subscription pricing. Based upon some parameters the service providers charge their customers. The other cost models that might be used by providers include effort based pricing, packaged and server based licensing, software renting, pay-per-use, utility based charging, revenue sharing with partners, freemium and advertisement-based models[25][26].

Pay-per-use charges customer periodically according to units used. The unit has fixed price set by provider. The customer does not need to sign the contracts to use the

software or applications. Also for customer, it's easy to switch to other provider with lower pricing.

In software rental model, the customer needs to pay for the negotiated subscription fee for certain limited time period. The rental price for service usage may depend upon the number of users in the consumer enterprise, length of the agreement, functionality of the SaaS service or size of the company. Customers have the benefits of more price negotiations. There are possibilities offers for varying usage terms and contract length. The total service cost is contractually defined and is predictable, also no hidden costs related to service are there. The provider may gain more revenue than in case of pay-per-use.

Most of the existing Cloud cost models mainly predict cost focussing only on IaaS parameters such as storage, computations, etc. The SaaS services are characterized by service parameters such as performance, response timeliness, usability, availability, scalability, reliability, customization, configurability, portability and so on. In order to determine the cost for the SaaS service these parameters must be considered. So a cost model is needed that would focus more on service parameters rather than infrastructure parameters.

CHAPTER- 2
LITERATURE SURVEY

The cost model determines what will service provider get from customer for providing services. The pricing can be fixed or dynamic. The fixed pricing charges the customer the same amount all the time whereas the dynamic pricing charges the customer amount that changes dynamically or it is market dependent, where the charges is based on real time market conditions[4].

In order to calculate the cost, the factors that affect the cost need to be discovered. These factors can be hardware, power and cooling infrastructure, software licensing, hardware and software maintenance, network bandwidth requirements, memory, storage, processing power usage, usage duration, allocation capacity, training, etc[5]

Different service providers apply different strategies or models for finding the cost. The most commonly used model is pay-as-you-go model[6] in which customers are charged with fixed price per unit. Many leading enterprises such as Amazon, Google App Engine and Windows Azure implement pay-as-you-go model. For SaaS pricing, the subscription based cost model[7]is widely used. The users are charged periodically for the services provided. Although, it does not determine cost according to what was being used by user.

2.1 Pricing Models and strategies

Many theoretical studies have been introduced in cloud computing for calculating the cost. These are as follows:

2.1.1 Genetic Pricing Model: A genetic algorithm for pricing was proposed that handles the problem of competitive price offers in service negotiations in the cloud computing markets. In this approach, suitable price is evolved from a pricing function. This genetic algorithm can be used with huge set of parameters and all the invalid parameters are discarded at later stage thus making the model work in complex and chaotic environments. This approach results in selecting the flexible provider in unpredictable market rather than the rigid provider[8].

2.1.2 Pricing as a Service: The static pricing strategy and different cost models for SaaS, PaaS, IaaS are determined[9]. The pricing model for SaaS users who have SaaS requirement for the short duration are charged more than frequent users as given below

$$Price_{SaaS} = BasePrice + \frac{R_{oc}+R_{re}*(1-p)*Q}{R_{tot}} \quad (1)$$

Where market conditions sets the *BasePrice*, R_{oc} is occupied resources, R_{re} is the reserved resources that can be booked by certain users and can also be cancelled before they are actually used; R_{tot} is the total amount of resources; p is the probability for the concealment of reserved resources; Q is a factor that indicates the scale of the current market. The pricing model for frequent users that have long term requirement can be given as

$$Price_{SaaSB} = Price_{SaaS} - \frac{R_{tot}*k_1*time+k_2*no.}{R_{tot}} \quad (2)$$

Where $Price_{SaaS}$ is price for short term users, R_{tot} is total amount of resources, *time* is duration for which user will occupy certain resources, k_1 and k_2 are time factor and amount factor respectively.

2.1.3 Cloud Storage Pricing: The cost model for cloud storage that considers the systems design access cost, usage cost, variable cost, discount cost and compensation cost[10]. Therefore the total cost of a user in a given period of time is given as

$$C^{ij} = C_a^{ij} + C_u^{ij} + C_f^{ij} - C_p^{ij} - C_b^{ij} \quad (3)$$

Where C^{ij} is total cost, C_a^{ij} is access cost, C_u^{ij} is usage cost, C_f^{ij} is variable cost , C_p^{ij} is discount cost and C_b^{ij} is compensation cost and i and j are the user level and service model respectively.There are different cost service model since different users have different storage needs. One is cost by storage space is for those users who large storage space need but do not need large bandwidth. Next is cost by time occupied is for those users who primarily pay attention to time their files but not need higher bandwidth and storage space. Next is comprehensive billing is for general users who require high bandwidth and large storage space but does not need higher using frequency. The proposed billing strategy for cloud storage such that user can pay only for resource usage while taking into account benefits of both providers and users. The billing strategy is developed on the basis of maximizing the provider's and user's

revenue by game theory.

2.1.4 Reserved, Spot and On-Demand instances: Amazon EC2 provides three different pricing model to customer: Reserved Instances, Spot Instances, On-Demand Instances[11].Reserved Instance charges low amount that is paid once and offers significant discount on hourly charge for that instance[12]. Three types of reserved instances i.e. light, medium and heavy utilisation reserved instances are available for customer. Spot Instance allows the customers to bid for unused capacity. The spot instances are charged with the spot price which is laid by Amazon. This price fluctuates based upon the demand and supply of the spot instance.
On-demand instance allow the customer to pay a fixed amount by hour with no commitments or upfront payments. The customers are free from cost of planning, purchasing and maintaining.

2.1.5 Pricing Model of cloud computing service: The cost model was proposed for cloud computing services that models multistage game with the cloud service provider who provide both reserved service and interruptible spot service[13]. The reserved service guarantees quality and is purchased at fixed price in the beginning. Whereas the spot service is can be interrupted by provider and has the price dynamic adjustment. Customer can choose whether they want to purchase quality reserve price in first stage or they can choose number of jobs to submit to reserve service and to spot service. The model is build that uses mix strategy of reserved service and spot service.

2.1.5 Pay as you go in the cloud: one watt at a time: Pay-as-you-go charges according to the actual use, independent of the time factor [14]. A per-watt-hour model is proposed which divides the costs among similar VMs and the less active VMs are freed from having to pay for others.

2.1.6 A Novel financial economic model: The cost model [15] for resources applies financial option theory to calculate the cost for the cloud computing commodities. The lower bound on the cost is given by finance model and upper bound is by compounded-Moores law. The model considers five parameters and maps them to BSM (Black-Scholes-Merton) Model to price the cloud resource. The five parameters

are initial investment, contract time, Rate of depreciation, quality of service and age of resources. The BSM formula for call option is given by

$$C(S,t) = N(d_1) * S - N(d_2) * K * e^{-r(T-t)} \quad (8)$$

Where $d_1 = \frac{\ln(S/K) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{(T-t)}}$ (9)

$$d_2 = d_1 - \sigma\sqrt{(T-t)} \quad (10)$$

The Black-Scholes formula for put option is

$$P(S,t) = N(-d_2) \times K \times e^{-r(T-t)} - N(-d_1) \times S \quad (11)$$

Where S is underlying asset price, K is strike price in contract, r is interest rate, σ is volatility, t is time and T is maturity time. $N(d)$ refers to normal distribution function on d. The mapping of parameters to BSM is as S is total investment, K is strike estimate, r is quality of service, and t is contract time and σ is volatility estimate.

2.1.7 Costing cloud computing services: TCO approach: The cost model was given that determine the TCO(total cost of ownership) for a cloud computing services[16]. The cost factors and its related cost type discovered form the basis for calculating cost. The cost types were then mathematically modelled to determine the TCO. For each cost factor, its cost types are added to determine the cost of that factor. The different cost factors that are identified are strategic decision, selection of cloud computing services and cloud types; evaluation and selection of service provider; service charge for IaaS, PaaS, SaaS; implementation, configuration, integration and migration; support; initial and permanent training; maintenance and modification; system failure; backsourcing and discarding.

2.1.8 Dynamic pricing in federated cloud: The dynamic pricing was introduced for federated clouds that allow different service providers to share resources[16]. The users of federated cloud can buy or sell the resources. In case of fixed pricing, when the market demand was high, the seller would not be able to raise his price thus lead to minimize seller welfare. Also with lower market demand, user would be charged more than the market prices thus lead to minimize user utility. For these situations, it is favourable to use dynamic pricing scheme such that the price is laid based on the levels of supply and demand.

2.1.9 Multi-service QoS based cost model: .The inter-organisation economic models[17] proposed for pricing the cloud network services when many cloud providers coexist in market that services the single application type. These models are capable of deriving the optimal resource provisioning in cloud networks. The QoS levels and Nash equilibrium price derives the optimal customer demand in given period of time according to maximizing individual cloud provider profits under competition.

2.1.10 Tiered pricing: In tiered pricing the services are offered in many tiers[18]. Every tier provides fixed computing and SLA with fixed price per unit. This type of pricing is adopted by Amazon cloud systems.

2.1.11 Cloud Option pricing: Cloud option pricing[19] uses financial option theory to mitigate risk and minimize the cost for the customers. The concept of option pricing scheme is to value an option, one must form a self-financing hedging strategy that replicates the pay-off of the option. Through an option holder can exercise the option and investor pays to purchase an option contract.

2.1.12 Optimal service pricing: A price demand model[20] proposed for cloud cache in data management services and for queries that executed in cache, a dynamic pricing scheme was given. The method applied estimates the correlations of cache service. The proposed approach provides long term profit maximization along with dynamic calibration to applications behaviour. This approach is for improving the optimization in data management service of cloud and can be extended to use as pricing strategy.

2.1.13 Agreement based pricing: Agreement based pricing enables businesses to adjust price, promotions, accelerators, multi-axis pricing and discounts, etc. These alterations can easily incorporate and gives more opportunities to grow the business. The enterprise internal management requirements are support by monetization option of business model that rapidly models fee allocation, account hierarchy alteration and automotive repeated processes. The agreement billing once placed can be easily integrated with existing business applications. It well organizes the end to end order to cash process and is adaptable to modifications in business model[21].

2.1.14 Variable pricing: the pricing is based on demand of particular resource type. For example, daytime pricing versus night time, weekends versus weekdays and so on. In order to reduce the overall pricing, different cluster nodes change their location dynamically[22].

2.1.15 cloud computing resources pricing algorithm: The algorithm is based on cloud bank agent model as a resource agency. It iteratively calculates the price. It analyzes the historical utilization ratio of resources, iterates the current price, access the resources available for next round and then determine the result price[23].

2.1.16 Hierarchical cloud pricing system

The cost model given in [24] is based on M/M/c queuing system. The model describes the utility functions for both consumers and service providers. The system architecture is proposed in which the distribution of users' task is done through the agent. The users are charged according to queues consumed by them. The queues can be schedule queue, computation queue and transmission queue. The utility function helps the providers to catch up the agreeable price through which they can gain advantage in the market. The utility functions for users help them to negotiate with providers to get cost effective services.

CHAPTER- 3

PROBLEM STATEMENT

The SaaS model lets the user utilize the software applications available over the cloud infrastructure. SaaS model is best suitable for the condition wherein the user doesn't wish or is unable to run the application on his system.

Consider a situation where in an organisation has reliant entities—such as authorised retailers or resellers —with which it has a well-built business relationship, but has a very basic IT process automation and information transfer, may be because of some technical or physical or financial constraints. In this scenario, by adopting a SaaS Model, the organisation will benefit as the SaaS will deliver a better IT process automation and information transfer. This results in revenue to the SaaS provider by the Organisation. This case leads us to the question "How much the payment should be?". Payment should be such which benefits both the End-User as well as the Service Provider. Payment is the cost plus the remuneration of the service provider. The next question then arises is "How to assess the cost?". Most of the existing Cloud cost models mainly predict cost focussing only on IaaS cost and as there is no separate cost model for SaaS, the SaaS providers are forced to use the existing Cloud cost model for predicting cost for their services and the Cloud cost model is unable to acknowledge and accommodate costs all the efforts made in developing the SaaS service. Therefore, a cost model is needed that would mainly focus on parameters related to SaaS services. These parameters may be configurability, customization, QoS parameters such as scalability, availability, usability, response timeliness and performance, portability, total cost of ownership and return on investment.

The problem can be formally defined as "to determine the cost that provider need to charge the end-user for a given SaaS service, such that it is reasonable to end-user and maximizes the provider's revenue and would mainly focus on service parameters".

CHAPTER- 4
IMPLEMENTATION

4.1 Proposed Approach

The preliminary assumption for the SaaS cost model construction is that the infrastructure is already available for service deployment.

In order to predict the cost, firstly we need to determine the different factors considered by provider or customer. These are as follows:

- Infrastructure required to deploy the service
- QoS parameters such as service availability, performance, usability, response timeliness, reliability and scalability.
    - Service availability means service is operational and accessible to customer when he wants to use it.
    - Performance means service is delivered with minimum response time and maximum throughput.
    - Response timeliness defined as ability to quickly solve the problems and quickly respond to the customer in case of service failure.
    - Scalability means the ability to handle increasing load by the service provider.
    - Usability means ability of service to be used by specified users to achieve specific goals efficiently.
- Configurability of the service. This aspect consists user interface, data, organizational structure, workflow and business logic. The customer can opt to configure any of these.
- Customization of the service. This allow customer to make changes to the service. The changes may be done to the functions or look or feel of the service. Some provider do not provide customizable SaaS, the user can only utilize the service.
- Security including data security, network security, application security and management security.

- Interoperatability that is the ability of the service to integrate the other information, such as information stored at other service provider, with the local information.
- Application isolation is whether the application is isolated by single instance-single user, multi instance, multi instance-multi user and virtualized SaaS.
- Data isolation is ability of service to isolate the data of the customer. The data isolation can be provided by dedicated database, shared database, and shared schema.
- Software fault tolerance
- SLA is service agreement between customer and service provider. It can include service definitions, duties of customer, performance measurement, help desk response time for various problems and so on.
- Portability is ability to migrate customer's data from one provider to another without any problem.
- TCO(total cost of ownership) includes license fee, support fee and user training fee
- ROI(return on investment) is benefits to provider in investment.

These factors are grouped to form different cost types. The different cost types and its related factors are given in table 4.1

Cost types	**Cost factors**
IaaS Cost(i)	Total investment of infrastructure(inf)
QoS(Q)	Quality variables(β),Service charges(p)
Storage cost(st)	Usage cost(u), variable cost(var)

Table 4.1 Cost types and its related cost factors

IaaS cost: IaaS cost is the amount that service provider charges for the total investment in setting up the infrastructure for providing the service. The price can be the existing infrastructure or the leased infrastructure. The existing infrastructure cost includes the charges which service provider is paying for its maintenance.

QoS cost: The QoS cost is amount charged for the service that is being provided by the provider. The QoS cost is that the provider charges for quality parameters of the service. The cost factors such as QoS parameters, customization, configuration,

portability, interoperatability, data isolation, application isolation and security forms the QoS cost.

Storage Cost: The provider charges the user for its storage space.

4.1.1 Basic Mathematical Modelling of Cost types:

For mathematical modelling of the cost, suppose cost type z Є Z is subject to set T={i,Q,s} and the cost factor v ЄV is subject to set V ={inf,p,u,var}. The different cost types can be determined as

Total investment of infrastructure: The cost factor the total investment of infrastructure inf determines the IaaS cost. The service provider can charge the user very small percent of its total investment. The IaaS cost C_i is given by

$$C_i = x\%\ of\ inf \quad \text{(i)}$$

Where x is the percentage that is specified by service provider and *inf* is the total investment of the service provider in laying the infrastructure for providing and deploying the services.

Quality variables: The various performance and quality cost factors such as security metrics, QoS metrics and software quality metrics can determine the QoS cost. These metrics can include customer, application, data and network security, service availability, usability, performance, response timeliness, SLA management, configurable UI, data and business logic of SaaS service, data isolation, and application isolation at virtualized SaaS. For performance based services and the economical based services the QoS cost is calculated differently. If the user opt for performance based service then the QoS cost is given by

$$C_Q = \sum_{i=1}^{n}(\beta_1 + \beta_2 + \cdots + \beta_n) * p \quad \text{(ii)}$$

Where β_i the quality variables are whose values are according to the quality parameters such as accuracy, response time, security etc; n is the number of quality parameters and p is the service charges.

For economical based services, no quality parameters are considered which only provides basic function of SaaS services. This basic level of SaaS meets the following metrics:

- All basic features of SaaS services including network security and management security.
- Multi-tenancy
- Service is non-configurable.

The QoS cost can be given as

$$C_Q = \beta * p \tag{iii}$$

Where β the service constant and p is the service charges

Usage cost: The usage cost is the actual amount which is charged for the actual use of the storage. The usage cost C_u is given by

$$C_u = C_{base} + S_u * P_u \tag{iv}$$

Where C_{base} is the base price, S_u is the actual storage space used and P_u is the usage price.

Variable Cost: The variable cost is the additional charges for excess part. If the storage space used is greater than the maximum storage space then there are charges for the excess storage space. The variable cost is given by

$$C_{var} = P_{var} * r \tag{v}$$

Where P_{var} is the variable price and r is the difference in actual measured storage space and maximum storage space (in GB).

The Usage cost and the variable cost when sum up together form the storage cost. Hence is given as

$$C_{st} = C_u + C_{var} \tag{vi}$$

The sum of IaaS cost, QoS cost and storage cost form the total cost given as

$$C_{tot} = C_i + C_Q + C_{st} \tag{vii}$$

The pricing schemes of SaaS are divided into the categories:

1 Free of charge: The users are not charged for the trial version of the service which is used for short period of time.
2 Monthly charge: The small business user can register for monthly pricing scheme.
3 Yearly charge: The regular business users can register for the yearly pricing. The discount is given to these users based on the ideal time period. Also compensation cost is subtracted when there is not satisfactory service delivery. So the cost for these customers comes out to be

$$C_{tot} = C_i + C_Q + C_{st} - \mathrm{C_d} - \mathrm{C_f} \quad \text{(viii)}$$

Where C_d is discount price and C_f is compensation price

Another option for pricing schemes for SaaS is divided into categories on the basis of the quality metrics of service.

1. Basic SaaS: This scheme provides all the basic features of SaaS. The following features are provided:
    - Service offered is multi-tenant.
    - Network security and management security is provided.
    - Service can be non-configurable or only UI is configurable.
    - Data is isolated through dedicated database.
    - Application is isolated through single instance – single user or multi instance.

2. Standard SaaS: The following features are provided in addition with basic SaaS features
    - QoS parameters such as scalability, usability, availability, performance, response timeliness.
    - Service level agreement contract can be made between customer and service provider.
    - The user can configure UI and data.
    - Data security is provided.
    - Data is isolated through dedicated database or shared schema.

- Application is isolated through multi-instance or multi instance- multi user.

3. Optimized SaaS: Optimized SaaS scheme provide following features in addition with standard SaaS features:
    - Application security is provided.
    - Fault tolerance software is provided.
    - Service with configurable UI, data and business logic is provided.
    - Data isolated through shared schema.
    - Application is isolated through multi instance – multi-user.

4. Integrated SaaS: Along with optimised features, it provide the following features:
    - Service can be customized.
    - Service with fully configurable including UI, data, business logic and workflow is offered.
    - Portability is provided.
    - Collaboration and sharing of information with other providers.
    - Data isolated through shared schema.
    - Application is isolated through virtualized SaaS.

4.1.2 Proposed Algorithm

Input: S_u actual storage space, performance based service or economical based service.

Output: Total Cost

I. Calculate IaaS cost C_i

$$C_i = \frac{x}{100} * inf$$

II. Calculate QoS cost

a. If performance based service then do

$$C_Q = \sum_{i=1}^{n} (\beta_1 + \beta_2 + \cdots + \beta_n) * p$$

b. If economical service then do

$$C_Q = \beta * p$$

III. Calculate storage cost

a. Calculate usage cost

$$C_u = C_{base} + S_u * P_u$$

If actual storage space is greater than maximum storage then

b. Calculate variable cost

$$C_{var} = P_{var} * r$$

c. Calculate storage cost

$$C_{st} = C_u + C_{var}$$

IV. Calculate total cost

$$C_{tot} = C_i + C_Q + C_{st}$$

4.2 Implementation

The proposed cost model is implemented by developing a cost forecasting tool. A service provider planning for delivering a SaaS service can estimate the total cost owned by him through the forecasting tool.

Technology: Netbeans IDE

In order to calculate the cost for the customer has following options

Server: The provider can select server from given different type of servers for service deployment he want to provide. The number of servers to be deployed can also be selected by customer.

Database: Also the type of database can be selected by provider as required.

Storage: The provider can estimate the cost for how much he required the storage space.

Subscription type: Which type of subscription he want to offer to customer.

.

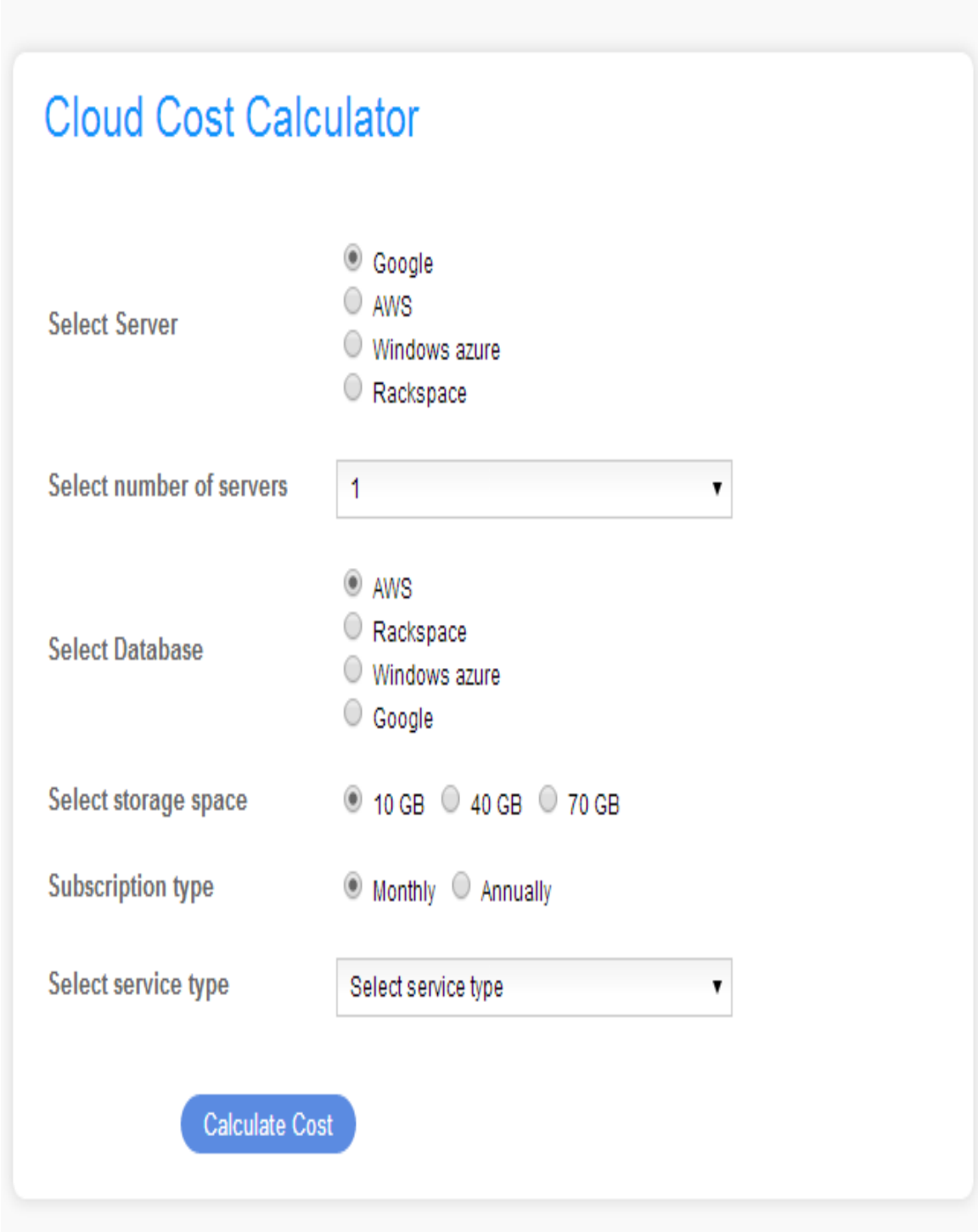

Figure 4.1 Cost calculating tool

Service type: Based on which type of service the customer wants, he can choose the service type. The customer can opt for economical based service, standard optimal and integrated service.

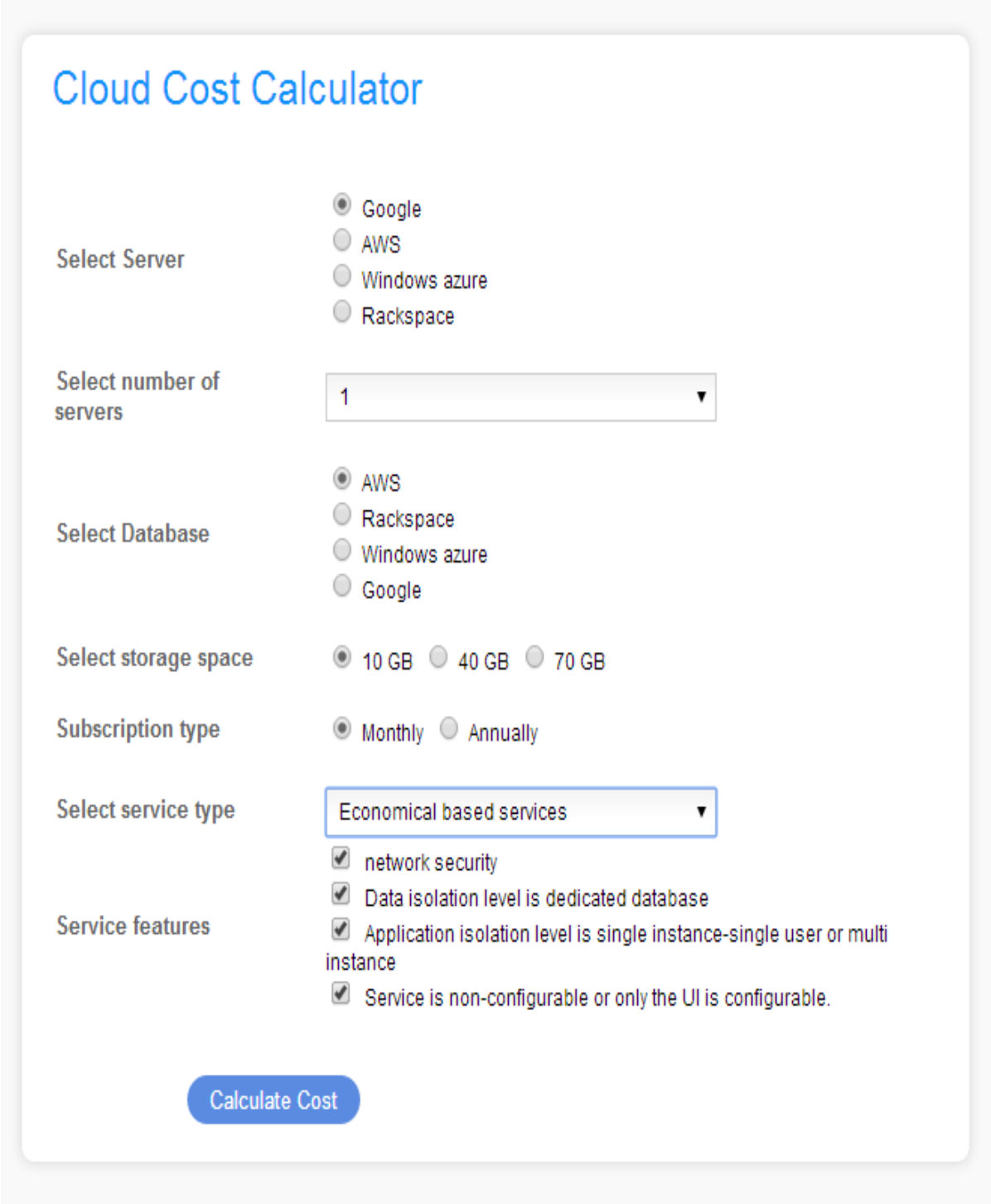

Figure 4.2 options shown for economical based services

The economical services include features network security, data isolation at dedicated database, service UI is configurable, application isolation at single instance single user. The service provider can select any of these features

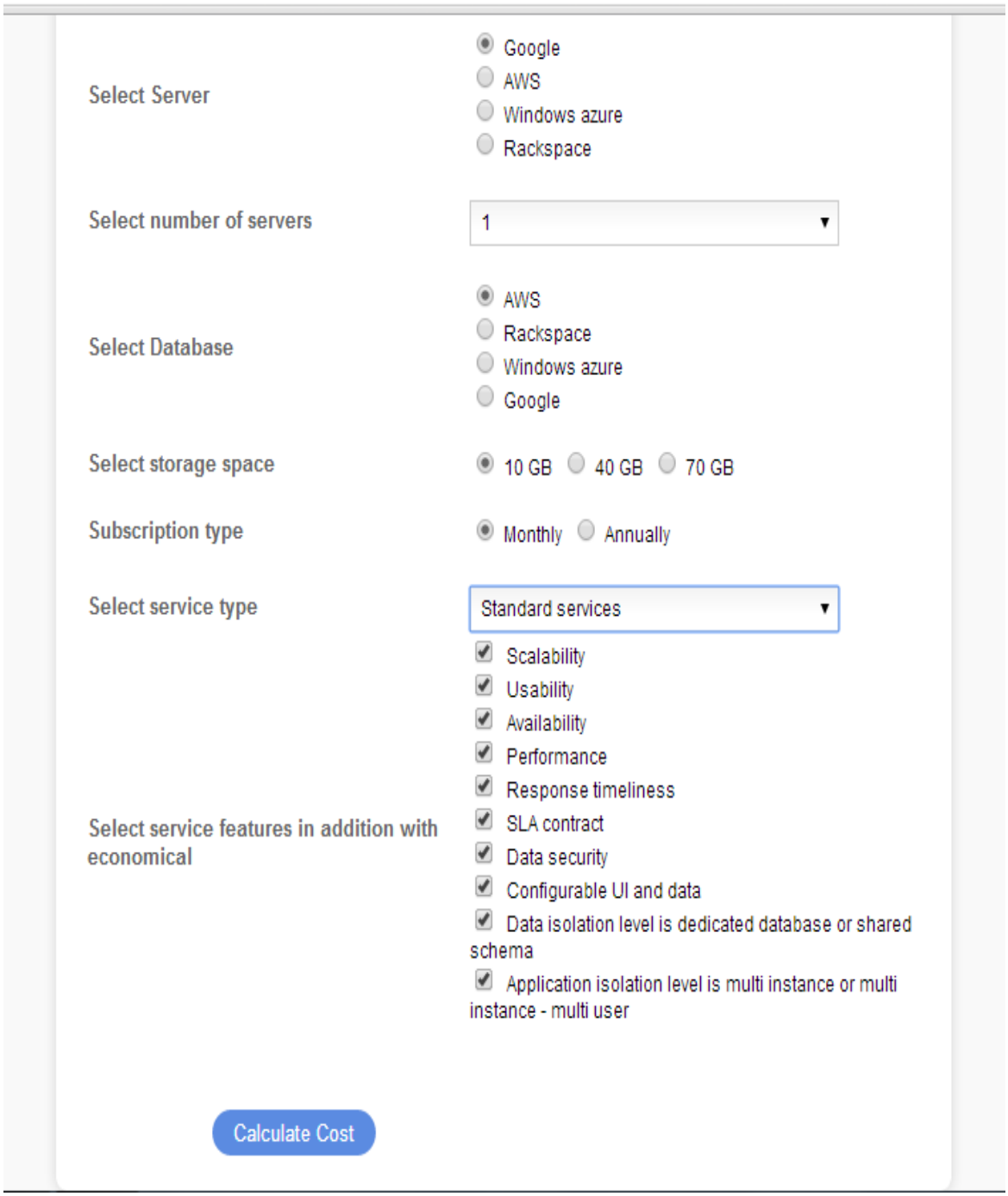

Figure 4.3 Options shown for standard services

The standard service include features such as scalability, usability, availability, reliability, performance, response timeliness, SLA contract, data security, configurable UI and data and so on. These features are provided in addition to economical service features. The provider can select any of these which he wants to provide for its service.

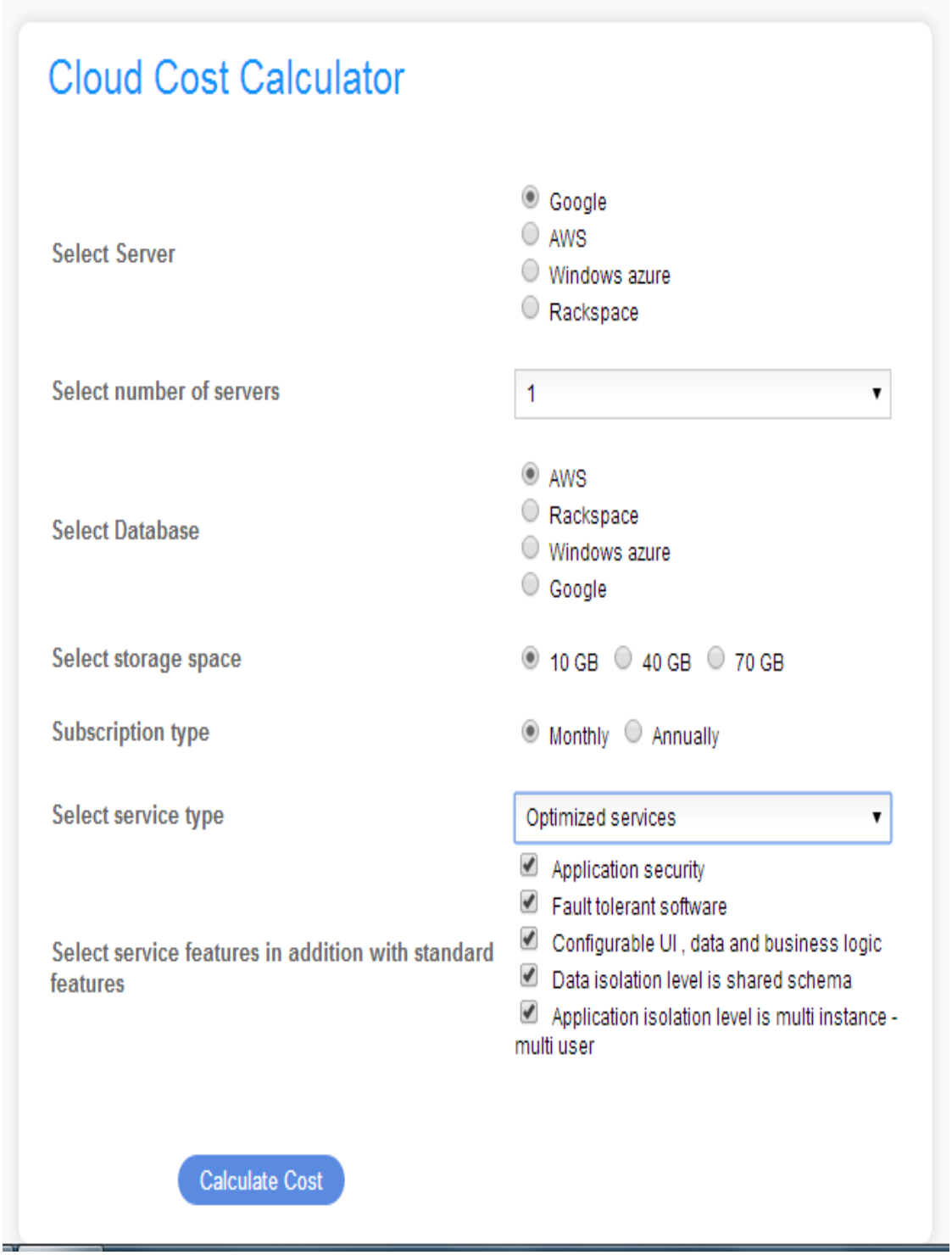

Figure 4.4 Options shown for Optimized services

The optimized service provide features such as application security, fault tolerance, configurable UI, data and business logic, data isolation at shared schema and so on. These features are provided in addition with standard service features.

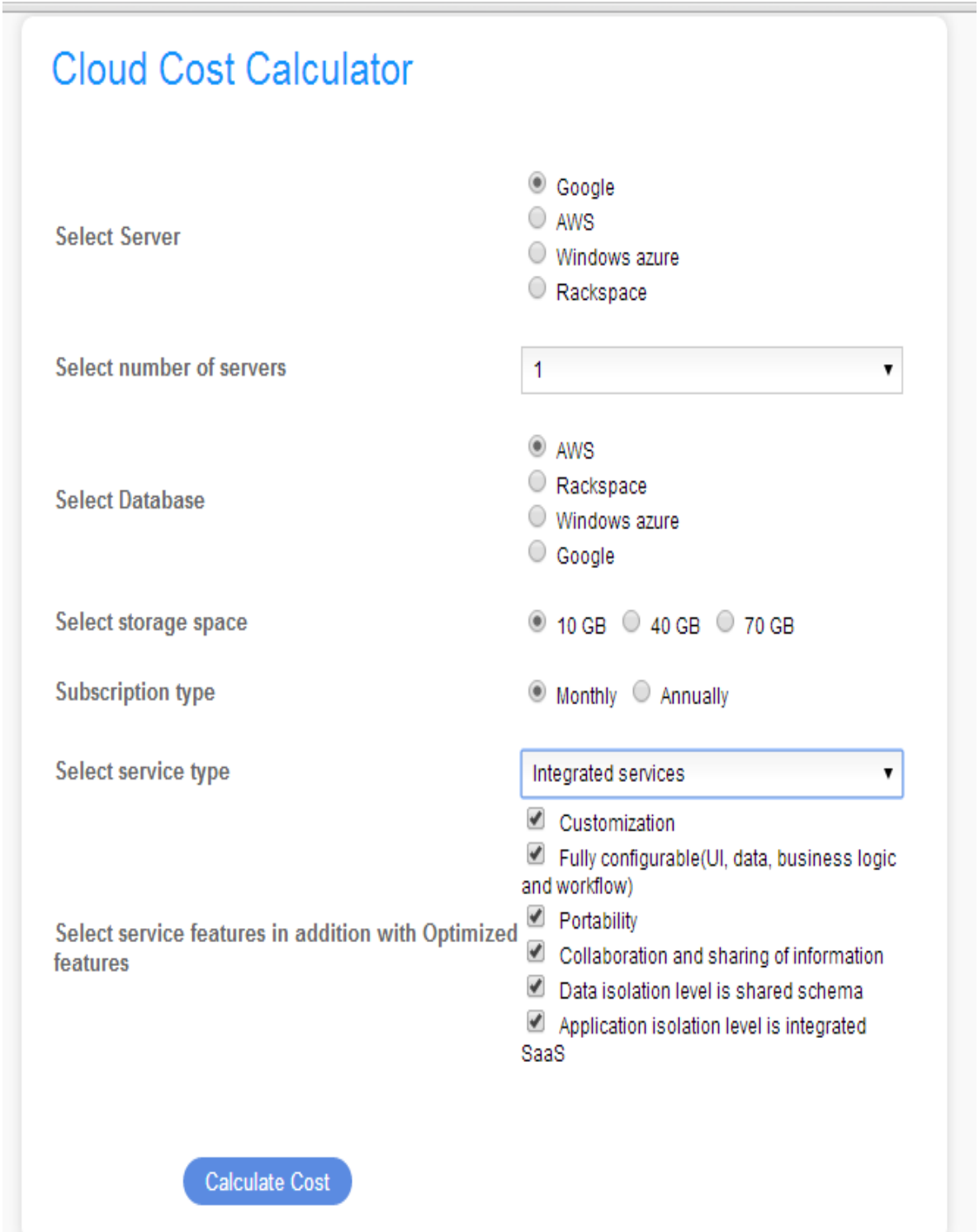

Figure 4.5 Options shown for integrated services

The Integrated services offer features such as customization, configurable data, UI, business logic and workflow, portability and so on. These features are offered in addition to optimized features.

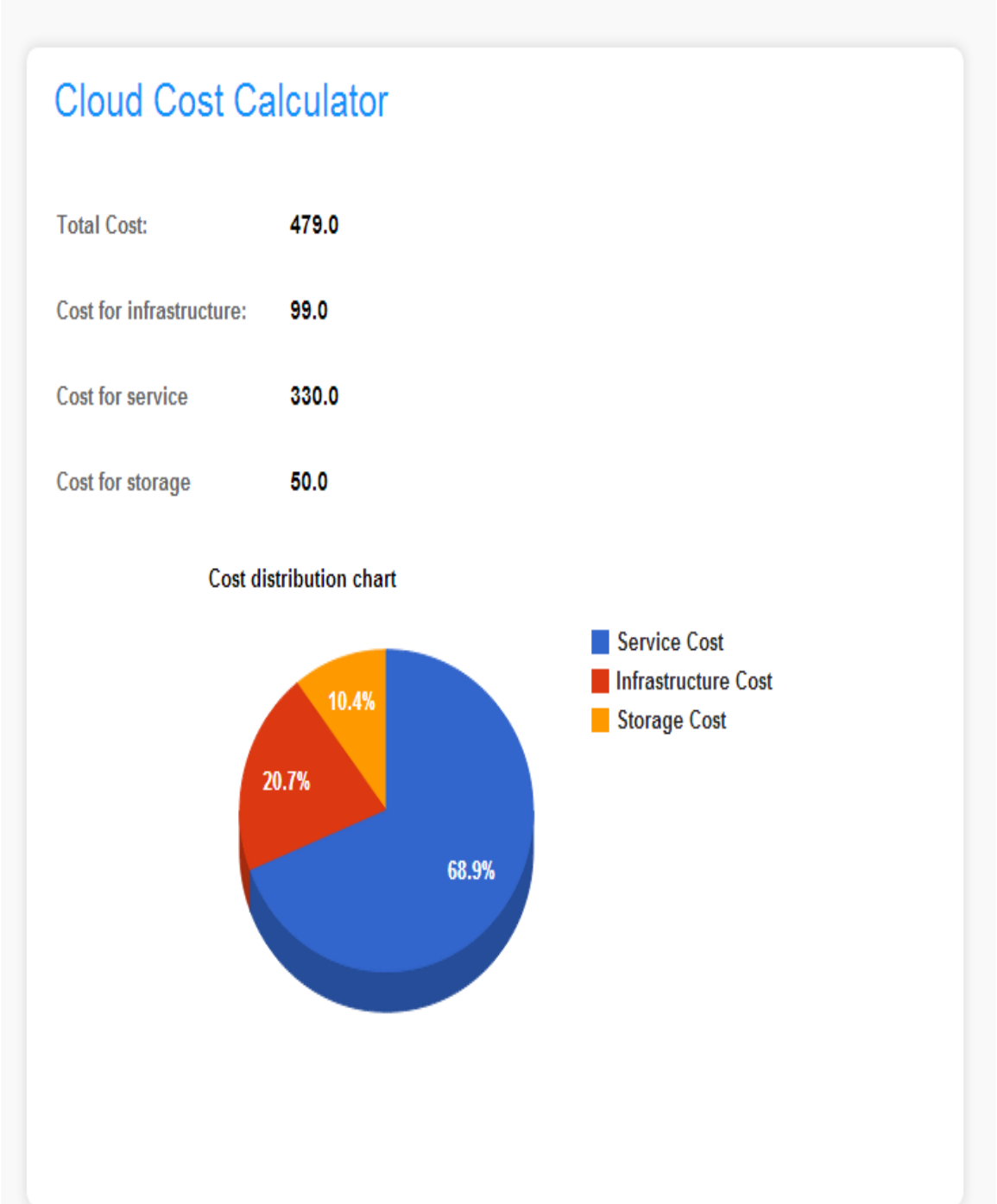

Figure 4.6 Calculated Cost by tool

Based upon the type of service selected, the total cost is determined by the forecasting tool.

CHAPTER- 5

ANALYSIS AND RESULTS

5.1 Cost distribution:

The total cost is determined by using following formula

$$C_{tot} = (x\%\ of\ inf) + (\sum_{i=1}^{n}(\beta_1 + \beta_2 + \cdots + \beta_n) * p) + (C_{base} + S_u * P_u)$$

The parameters values are assumed that are shown in table 5.1

Parameter	Parameter name	Parameter Value
x	Percentage to be taken	30
inf	Total infrastructure	330
C_{base}	Base price	0
S_u	Storage space	10
P_u	Usage price	5
p	Service charge	10

Table 5.1 Parameters and assumed value

The following are the formulas are used for finding the total cost for different type of service schemes:

a) Economical based service: the total cost calculated by using following formula

$$C_{tot} = (x\%\ of\ inf) + (\sum_{i=1}^{n=4}(\beta_1 + \beta_2 + \cdots + \beta_n) * p) + (C_{base} + S_u * P_u)$$

The following are values assumed for parameters of formula for find the total cost shown in table 5.2

Parameter	Parameters Name	Parameter Value
β_1	Configurability	7
β_2	Network Security	8
β_3	Application isolation	10
β_4	Data isolation	8

Table 5.2 Parameters and assumed values for Economical services

b) Standard Services: the total cost calculated by using following formula

$$C_{tot} = (x\%\ of\ inf) + (\sum_{i=1}^{n=14} (\beta_1 + \beta_2 + \cdots + \beta_n) * p) + (C_{base} + S_u * P_u)$$

The four parameters are economical based parameters and defined in table 5.2 and ten values and parameters shown in table 5.3

Parameter	Parameter name	Parameter Value
β_1	Scalability	4
β_2	Usability	3
β_3	Availability	3
β_4	Performance	5
β_5	Response timeliness	6
β_6	SLA contract	2
β_7	Data Security	5.5
β_8	Configurable UI and data	2
β_9	Data isolation	2
β_{10}	Application isolation	6

Table 5.3 Parameters and assumed values for Standard services

c) Optimized Service: the total cost calculated by using following formula

$$C_{tot} = (x\%\ of\ inf) + (\sum_{i=1}^{n=19} (\beta_1 + \beta_2 + \cdots + \beta_n) * p) + (C_{base} + S_u * P_u)$$

The four parameters are economical based parameters and defined in table 5.2 and ten values and parameters shown in table 5.3 and five values are shown in table 5.4

Parameter	Parameter name	Parameter value
β_1	Application security	10.5
β_2	Fault tolerant software	13.5
β_3	Configurability	14
β_4	Data isolation	2
β_5	Application isolation	5

Table 1.4 Parameters and values for Optimized services

d) Integrated service: the total cost calculated by using following formula

$$C_{tot} = (x\%\ of\ inf) + (\sum_{i=1}^{n=25} (\beta_1 + \beta_2 + \cdots + \beta_n) * p) + (C_{base} + S_u * P_u)$$

The four parameters are economical based parameters and defined in table 5.2 and ten values and parameters shown in table 5.3 and five values are shown in table 5.4 six parameters are defined in table 5.4

Parameter	Parameter name	Parameter value
β_1	Customization	15.5
β_2	Fully configurability	16
β_3	Portability	14
β_4	Collaboration and sharing of information	11
β_5	Data isolation	5
β_6	Application isolation	8

Table 5.5 Parameters and assumed values for Integrated services

The total cost obtained for different type of service is shown in table 5.6

Cases(Type of service)	Total Cost(in Rupees per month)
Economical based service(1)	479
Standard Service(2)	864
Optimized Service(3)	1314
Integrated service(4)	2009

Table 5.6 Total Cost according to service types

The cost distribution of different type of service is illustrated in figure showing the bar graph shown in figure 5.1 and the pie chart shown in figure 5.2.

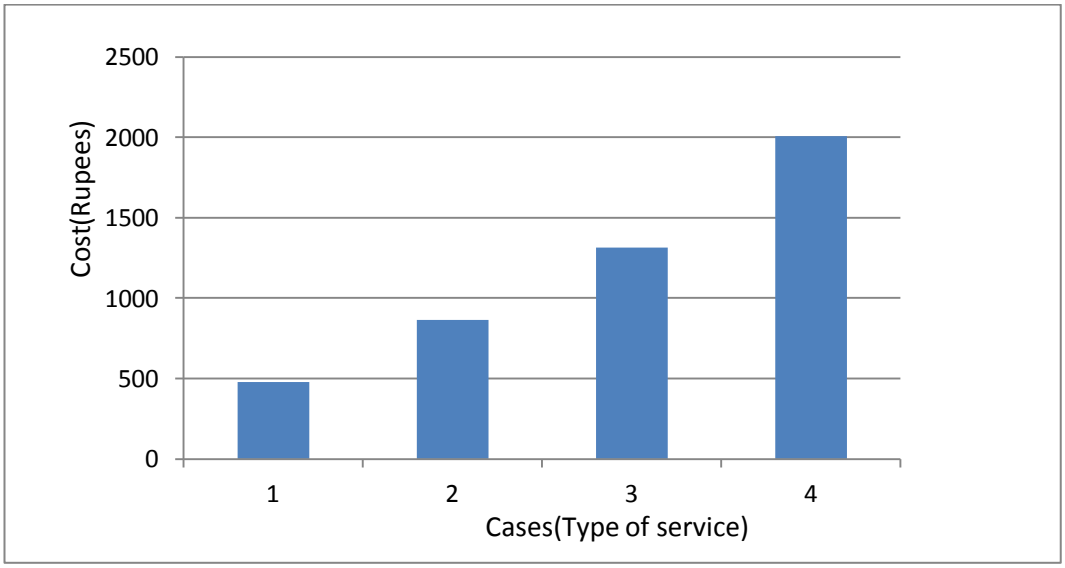

Figure 5.1 Bar graph showing cost for different type of services per month

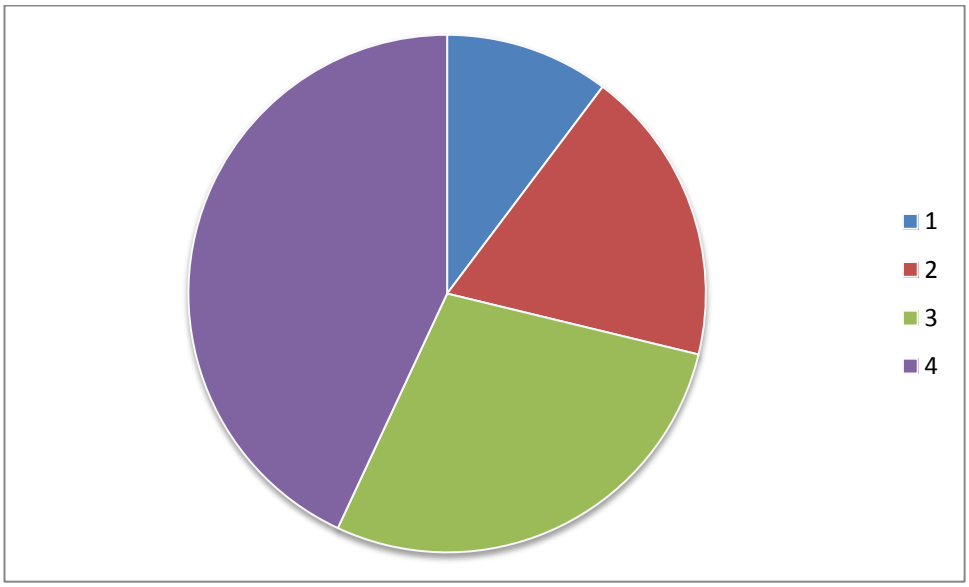

Figure 5.2 Pie chart showing cost distribution of type of service per month

5.2 Effect of Cost factors on Cost:

The standard services are assumed for evaluating the effect of different parameters. The assumed values for the parameters are taken from values shown in table 2, table 3 and table 4.

Performance:

For evaluating the effect, the performance parameter values are varied as shown in table 5.7.

Performance parameter values	**Total Cost(Rupees)**
17	984
25	1064
33	1144
40	1214
49	1304
56	1374

Table 5.7 Total cost corresponding to different performance parameter values

The graph shown in figure 5.3 shows the effect of performance factor on the total cost. The total cost increases with the increase in the performance factor.

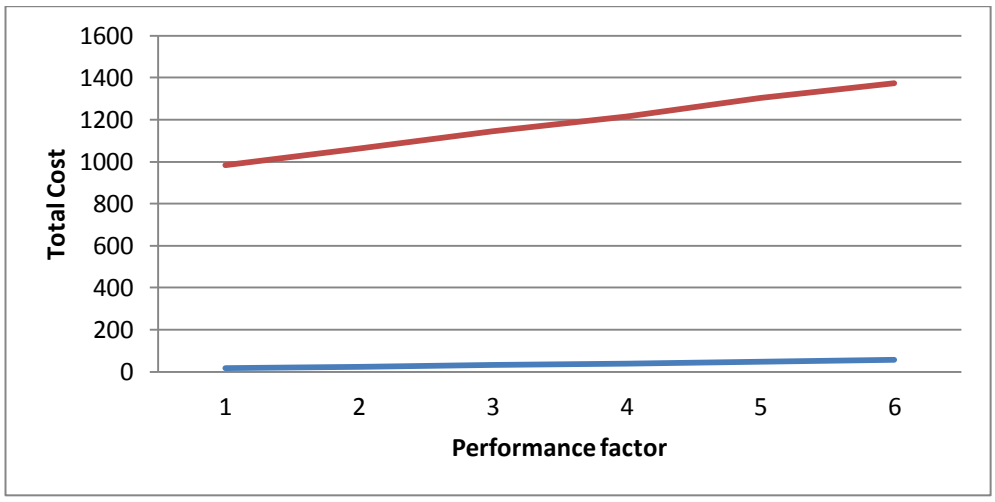

Figure 5.3 Effect of performance on total cost

Scalability:

For evaluating the effect, the scalability parameter values are varied as shown in table 5.8.

Scalability	Total Cost
8	904
19	1014
26	1084
35	1174
41	1234
55	1374

Table 5.8 Total cost corresponding to scalability parameter values

The graph has shown in figure 5.4 that shows the effect of scalability factor on the total cost. The total cost increases with the increase in the scalability factor.

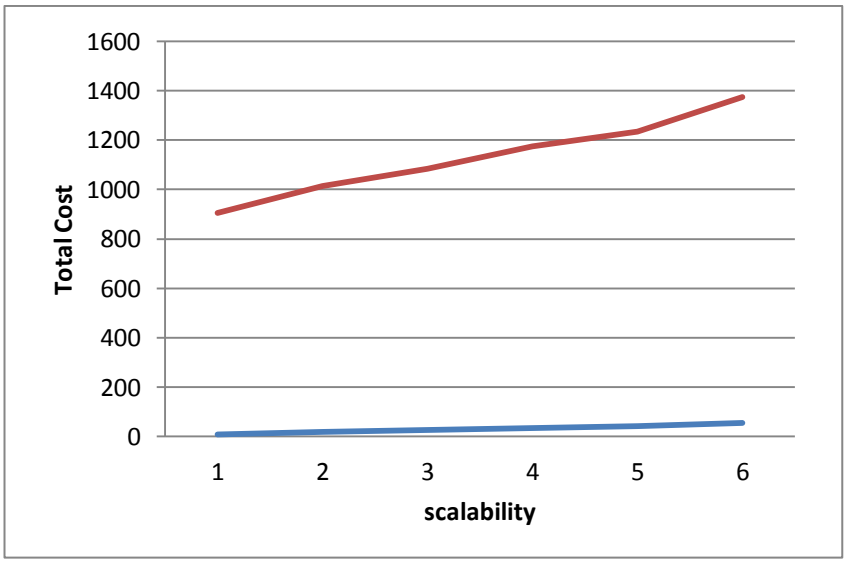

Figure 5.4 Effect of scalability on total cost

Availability:

For evaluating the effect, the availability parameter values are varied as shown in table 5.9.

Availability	Total Cost
3	864
15	984
23	1064
30	1134
37	1204
45	1284
53	1364

Table 5.9 Total cost corresponding to availability parameter values

The graph has shown in figure 5.5 that shows the effect of availability factor on the total cost. The total cost increases with the increase in the availability factor.

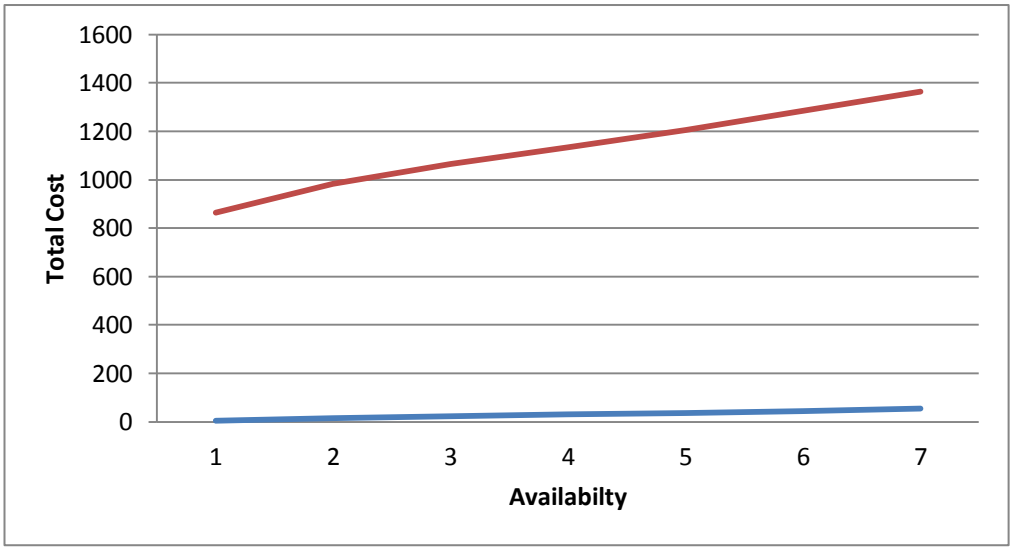

Figure 5.5 Effect of availabilty on Total Cost

Response timeliness:

For evaluating the effect, the response timeliness parameter values are varied as shown in table 5.10.

Response timeliness	Total Cost
20	1004
29	1094
37	1174

46	1264
53	1334
60	1404
69	1494

Table 5.102Total cost corresponding to response timeliness parameter values

The graph has shown in figure 5.6 that shows the effect of response timeliness factor on the total cost. The total cost increases with the increase in the response timeliness factor.

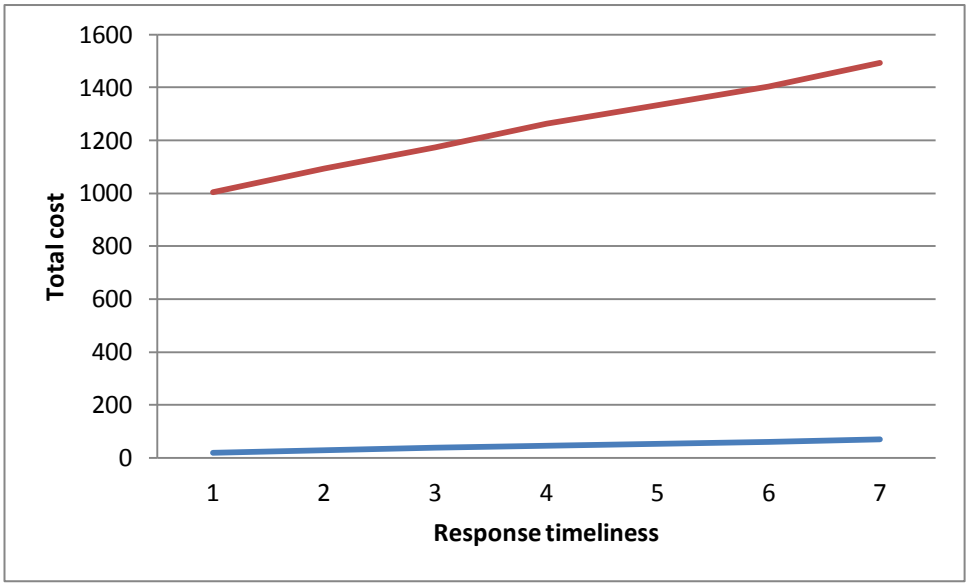

Figure 5.6 Effect of Response timeliness on total Cost

Data security:

For evaluating the effect, the data security parameter values are varied as shown in table 5.11.

Data security	**Total Cost**
5.5	864
12	929
21	1019
23	1099

37	1179
45	1259

Table 5.11 Total cost corresponding to data security parameter values

The graph shown in figure 5.7 shows the effect of data security factor on the total cost. The total cost increases with the increase in the data security factor.

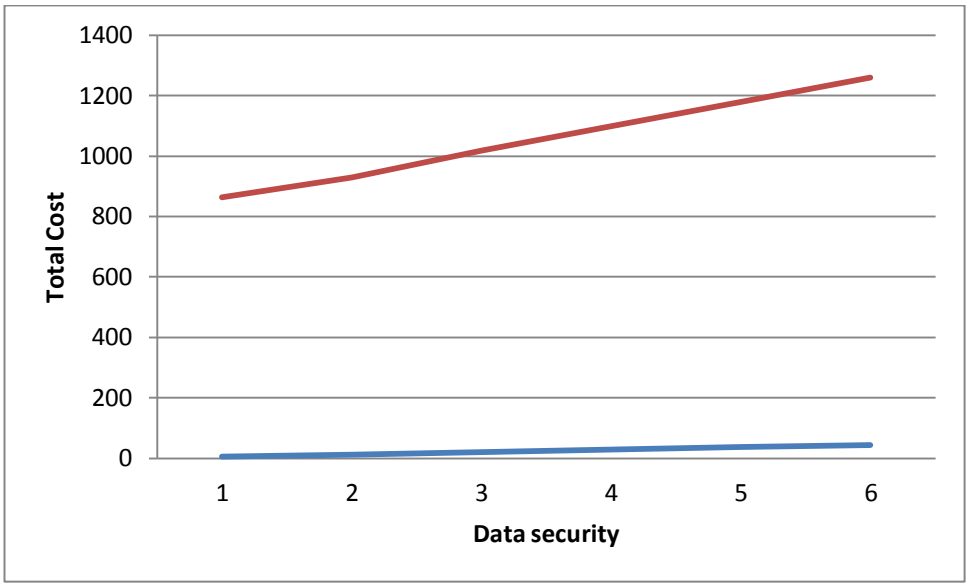

Figure 5.7 Effect of data security on Total Cost

CHAPTER- 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this thesis, the cost model has been proposed to predict the cost for the SaaS services. The cost model proposed calculates the total cost based on the cost factors for the SaaS services. These cost factors are the factors which are associated with different parameters of the SaaS and add-up to give us the total cost for the SaaS service. Such factors include infrastructure, configurability, customization, QoS parameters such as scalability, availability, usability, response timeliness and performance, portability, total cost of ownership and return on investment. These factors characterize the SaaS services. The proposed approach provides following benefits

- The effort is calculated effectively.
- SaaS service provider will be able to price its service as per consumer's requirement in form of various service packages.
- Using proposed model, service provider can easily append features in existing service and able to calculate the additional effort without recalculating the entire cost of application.

6.2 Future Scope

The proposed model can be further extended by adding more service related parameters. Also, the proposed cost model can be made dynamic by considering market related factors such as demand and supply.

REFERENCES

[1] R. Buyya , C. Vecchiola, S. T. Selvi, “Mastering cloud computing,” *Tata McGraw-Hill Education,* 2013.

[2] S. Subashini, V. Kavitha, “A survey on security issues in service delivery models of cloud computing”, *Journal of Network and Computer Applications,* vol. 34, no. 1, pp. 1-11, 2011.

[3] P. X. Wen,L. Don, “Quality Model for Evaluating SaaS Service”, *IEEE Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT),* IEEE, pp. 83-87, 2013.

[4] A. Osterwalder, “The business model ontology: A proposition in a design science approach”, *Ecole des Hautes Etudes Commerciales HEC* 173, Institut d’Informatique et Organisation. Lausanne, Switzerland, University of Lausanne, 2004.

[5] J. Kozhipurath, "Cloud Service Costing Challenges." *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* IEEE, 2012.

[6] M. Armbrust, A. Fox, R. Griffith et al., “A view of cloud computing,” *Communications of the ACM,* vol. 53, no. 4, pp. 50-58, 2010.

[7] L. Youseff, M. Butrico, D. Da Silva. "Toward a unified ontology of cloud computing." *Grid Computing Environments Workshop, GCE'08*, IEEE, 2008.

[8] M. Macias, J. Guitart, “ A genetic model for pricing in cloud computing markets,” *Proceedings of the 2011 ACM Symposium on Applied Computing*, TaiChung, Taiwan, pp. 113-118, Mar. 21-24, 2011.

[9] X. Wu, W. Zhang, D. Wanchun. "Pricing as a Service: Personalized Pricing Strategy in Cloud Computing." IEEE *12th International Conference on Computer and Information Technology (CIT)*, IEEE, pp. 1119-1124, 2012.

[10] H. Li, Y. Yang, B. Zheng. "Research on Billing Strategy of Cloud Storage." *IEEE Fourth International Conference on Multimedia Information Networking and Security (MINES),* IEEE, pp. 624-627, 2012.

[11] “Purchasing Options”, *Amazon Web Services* [Online], Available: http://aws.amazon.com/ec2/purchasing-options/, Accessed On: Jun. 23, 2014.

[12] A. Gohad, N. C. Narendra, and P. Ramachandran, “Cloud Pricing Models: A Survey and Position Paper,” *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* IEEE, pp. 1-8, 2013.

[13] D. Ma, J. Huang, “The pricing model of cloud computing services,” *Proceedings of the 14th Annual International Conference on Electronic Commerce,* ACM, pp. 263-269, 2012.

[14] K. Teramoto, H. H. Huang, “Pay as You go in the Cloud: One Watt at a Time,” *High Performance Computing, Networking, Storage and Analysis (SCC),* IEEE, pp. 1546-1547, 2012.

[15] B. Sharma,R. K. Thulasiram, P. Thulasiraman, “Pricing cloud compute commodities: a novel financial economic model,” *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012),* IEEE Computer Society, pp. 451-457, 2012.

[16] B. Martens, M. Walterbusch,F. Teuteberg. "Costing of cloud computing services: A total cost of ownership approach." *45th Hawaii International Conference on System Science (HICSS),*IEEE, pp. 1563-1572, 2012.

[17] M. Mihailescu,Y. M. Teo, “Dynamic resource pricing on federated clouds,” *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid),* IEEE,pp. 513-517, 2010.

[18] P. H. R. Pal, P. Hui, “Economic models for cloud service markets.” in ICDCN, pp. 382–396, 2012.

[19] L. Youseff, M. Butrico, D. Da Silva, “Toward a unified ontology of cloud computing” In *Grid Computing Environments Workshop GCE'08*, IEEE, pp. 1-10, Nov. 2008.

[20] M. R. Rahman, Y. Lu, I. Gupta, “Risk aware resource allocation for clouds,” [Online], Available: https://www.ideals.illinois.edu/handle/2142/25754, Accessed On: Jun. 23, 2014.

[21] V. Kantere, D. Debabrata, G. Francois et al. "Optimal service pricing for a cloud cache." *IEEE Transactions on Knowledge and Data* Engineering, vol. 23, no. 9, pp. 1345-1358, 2011.

[22] C. Preimesberger, "Cloud computing: Monetizing the cloud: 10 benefits of agreements-based billing." [Online]. Available: http://www.eweek.com/c/a/Cloud-Computing/Monetizing-the-Cloud-10-Benefits-of-AgreementsBased-Billing-384798/, Accessed On: Jun. 23, 2014.

[23] R. Moreno-Vozmediano, R. S. Montero, I. M. Llorente. "Multicloud deployment of computing clusters for loosely coupled MTC applications," *IEEE Transactions on Parallel and Distributed* Systems, vol. 22, no. 6, pp. 924-930, 2011.

[24] H. Li, J. Liu, G. Tang, "A pricing algorithm for cloud computing resources," *International Conference on Network Computing and Information Security (NCIS)*, IEEE, vol. 1, pp. 69-73, 2011.

[25] Z. Li, M. Li, "A hierarchical cloud pricing system," 203 *IEEE Ninth World Congress on Services (SERVICES)*, IEEE, pp. 403-411, 2013.

[26] L. M. Sainio, E. Marjakoski, "The logic of revenue logic? Strategic and operational levels of pricing in the context of software business," *Technovation*, vol. 29, no. 5, pp. 368-378, 2009.

[27] D. Ferrante, "Software Licensing Models: What's Out There?," *IT Professional*, vol. 8, no. 6, pp. 24-29, 2006.

[28] G. Kulkarni, S. Mandhare, D. Bendale, S. Belsare, N. Patil, "Software as service cloud", *2012 International Conference on Computer Science and Service System*, IEEE, pp. 442-445, 2012.

LIST OF PUBLICATIONS

Communicating Paper

1. Tript Kaur, Ashish Aggarwal, Damandeep Kaur, "Cost Model for SaaS services", submitted in *The 5th International Conference on Cloud Security and Big Data, Confluence-2014* held at Amity University, Uttar Pradesh. Submitted paper on 15th, May 2014.

2. Tript kaur, Ashish Aggarwal, Damandeep Kaur, "Cost Model for SaaS services", submitted in *IEEE Cloud Computing for Emerging Markets* to be held at 15-17 October 2014 in Banglore, India. Paper submitted on 14 July 2014.