Cost Aware Resource Selection in IaaS Clouds

Uzma Bibi

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology,
Islamabad, Pakistan

***Abstract*—One of the main challenges in cloud computing is to cope up with the selection of efficient resources in terms of cost. There are various cloud computing service providers which dynamically provide resources to the customers through different pricing policies. Based on the different APIs and pricing policies of the service providers, it becomes difficult for the customers to select the best service provider in terms of cost. In some cases, if the usage of the resources provided by a datacenter exceeds certain limit, then the providers cannot offer more resources to the customers as new VMs cannot be created. Hence, even if the customer chooses the best provider based on the least cost parameter, still there is no guarantee that the provider allocates complete resources to the customer. For this reason, I present system architecture that selects the best service provider based on the customer requirements mainly the cost. The proposed architecture also performs resource management by automatically providing new VMs from the available service providers in the inter cloud. The proposed system is based on five clouds i.e. Amazon EC2, Cloudsigma, Google, GoGrid, and Windows Azure. An interface is designed for obtaining the user requirements. These user requirements are matched with the design database of five cloud providers and based on the matched values; the catalog of optimal costs for each particular cloud is shown to the user. Then Cost Aware Resource Selection algorithm is run for determining the lowest optimal cost for Instance based approach and Quantity based approach. The algorithm tackles two domains of clouds for the algorithm i.e. Single Cloud and Multi Cloud.**

Keywords*—Cloud computing; pay-as-you-go; infrastructure as a service; cost aware resource selection; virtual machines; hypervisor; instance based approach; quantity based approach; single cloud; multi cloud

I. INTRODUCTION

A. Overview of Cloud Computing

Cloud computing is an emerging paradigm which provides the computing services as fifth utility [1]. Cloud computing provides the resources on pay as you go basis which gives new perspective and identity to the cloud business.

Clients and system administrators can deploy their applications and web services through the allocation of resources by cloud providers. They do not have to invest an upfront cost on Infrastructures and do not have to manage the resources on their own. The tasks are reduced and resources are all managed by the cloud providers providing the resources. The definition of cloud computing was represented by Vaquero [2] after reviewing many different other definitions of cloud computing:

"Clouds are a large pool of easily usable and accessible virtualized resources such as hardware, development platforms and/or services. These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs".

National Institute of Standards and Technology (NIST) has also presented the definition of Cloud computing that is easily understandable [3]:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models".

Service on demand, network access, elasticity, pooling of resources, and measured service are the five important characteristics present in the NIST definition of Cloud Computing [3]. There are deployment and service models in the cloud environment. The deployment models are public cloud, private cloud, community cloud, and hybrid cloud (the combination of both private and public clouds). The service models are SaaS (Software as a Service, PaaS (Platform as a Service), and IaaS (Infrastructure as a Service). Fig. 1 shows the service models and the deployment models in cloud computing. It summarizes the operation of three service models on the top of deployment models.

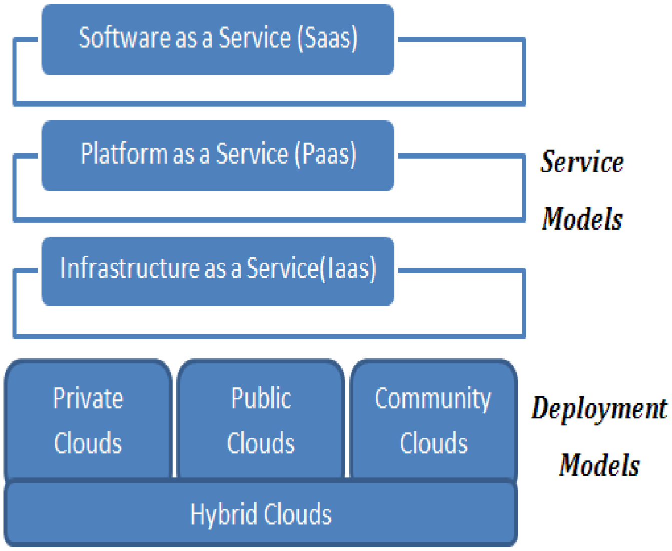

Software as a Service (Saas)
Platform as a Service (Paas)
Infrastructure as a Service(Iaas)
Service Models
Private Clouds
Public Clouds
Community Clouds
Hybrid Clouds
Deployment Models

Fig. 1. Service and Deployment Models in Cloud Computing (Source: [4]).

Cloud computing brings some features that are attractive to clients and can change the procedure of accessing the resources from clouds making it easy and convenient. The best feature among all features that cloud computing provides is usage of a service as a utility that is pay as you go model of accessing and using the resources, elasticity in resource provisioning, resource management, and having the infrastructure without investing on it. Broad network access is another feature of cloud computing that lets the clients access the resources from Internet [3, 5].

Section II provides the related work of other authors highlighting their limitations and strengths of my proposed work. Section III discusses the proposed framework and the underlying design. Section IV provides implementation and the experimentation results. Section V draws my conclusions.

B. Cost Aware Decision System for Resource Selection in Clouds

Earlier, there have been some systems presented for the system administrators to select resources and deploy their web applications and services on clouds. Decision system for resource selection [6] is one of such approaches. Through this approach the system administrators have an ease in selecting the best resource provider from all over the world.

The cost aware resource selection decision system maintains a database of various services provided by different cloud providers. We take five cloud providers for our system on general basis; these are Amazon EC2, Google, GoGrid, CloudSigma, and Windows Azure. The system obtains requirements from the clients and then a comparison is done for the various services saved in the database. Based on the services that match the requirements and charge the lowest cost are selected as the pertinent cloud provider for the client. The best resource can be selected from two or more than two cloud providers. Then the computation cost among the modules is evaluated and a comparison is done to suggest the best resource to the users.

The cost aware resource selection decision system is a solution to the problems that a system administrator faces during the resource selection that best matches their requirements. It is assumed that the resources of the applications that are needed to deploy them are already known to the system administrator. This research covers the main important factor i.e. cost. It provides the cost awareness for resource selection in clouds but it does not cover other parameters like performance and is left for further extension to this research.

II. Related Work

Zhang [7] worked on providing the cloud resources on user demands considering the cost factor as minimum as possible. The focus of the research was providing the network, storage, and computing resources. The author managed to work on three layers to provide the best solution. Although the approach was good in providing the cheapest resource provider, it did not provide the complete solution of getting the resources after having the knowledge of the best provider. The users had to manage the resource selection and retrieval at their own. Prashant and Upendra [8] propose a cost aware system that optimizes the selection of virtual server that minimizes the cost. The researchers also focused on the mechanisms to reduce the time in new configurations. Their main focus was to remove the tradeoff between cost and time for cloud resources in return ignoring the solution to provide the best resource provider offering minimum cost. My proposed system is complementary to this and provides the best cost awareness for efficient resource selection in inter clouds.

Liu and Zhang [9] presented a framework that chooses the resource provider based on the quality of service. The work comprised of discovery agents that managed the discovery of cloud resources using hash indexing. The whole network was divided into different domains to collect the Quality of Service information to choose the best resource provider. The focus was purely abstract covering the quality with the least focus on cost. My system removes the abstraction level, with a complete focus on minimizing the cost.

A survey of cloud computing presented in [10, 11] highlight the architecture and key principles about the resource selection with high performance and minimum cost. These survey papers presented the service selection algorithm that is adaptive to different kind of environments with scalability and availability. The research is only limited to design and architecture of cloud computing for resource selection covering performance and cost factors at a very abstract level. Due to insufficient resource available at times, the cloud provider may not be able to provide the demanded resources to the clients [12]. High cost of resources to meet the requested services by clients has led to the pay-as-you-go model to avoid the fluctuations in cloud computing. Seagull [12] was introduced as a solution to overcome the outburst in cloud computing due to scarcity of resources. This technique focused well on overcoming the sudden outburst but did not provide a unified solution to provide cost awareness to the clients for selecting resources.

Different cloud providers need to work on how to price the cloud resources requested by the clients from different environments [11, 13]. Enhanced ant colony system was introduced by Wang [13] which discussed the composition cost of data transfer for composing a service. A service selection algorithm was introduced which was able to deal with parallel relations between services. This algorithm only focused on the location of service providers and users, neglecting the other parameters which have a great impact on the cost evaluation. My system covers all the basic parameters to evaluate cost of the resources in an efficient way.

Inter cloud architectures and brokering systems for cost efficiency proposed by Nikoley and Rajkumar [14] discuss the inter cloud environments that facilitate the brokering of cloud resources in clouds. Kingfisher [15] is a model that presented the cost aware elasticity of cloud resources. This system focused on reducing the cost of virtual servers and their configurations. Kingfisher removed the tradeoff between cost and time but did not provide the efficient solution to the users to select the best service based on minimum cost. My system presents the unified solution to this problem. Mistral [16] addressed the tradeoff between three parameters i.e. power, performance, and cost. It is a framework that optimizes these three parameters to maximize the utility. Its focus is on addressing the tradeoff neglecting the parameter cost, which is a major factor in resource selection. Mansoor and Lakshman [17] proposed a resource allocation algorithm for distributed clouds. The objective was to allocate the resource by minimizing the distance between cloud providers and clients, optimizing the selection of servers in clouds. The performance of the algorithm was evaluated through simulations. As the proposed system was based on resource allocation, it did not provide a solution to the cost aware selection of resources. Whereas, my proposed system satisfies the solution to cost aware resource selection in inter clouds.

Meikang [18] proposed a two dynamically based resource allocation algorithms that adjust the updated task information for execution. These algorithms allocate the resources to the clients and schedule the order of execution for tasks. The objective of these algorithms was to improve the performance. OPTIMIS [19] was proposed for dynamic provisioning of cloud resources. The objective of the technique was to optimize the life cycle of a service including its construction, deployment, and operation on the basis of cost and other factors. It aimed at providing a reliable and trustful cloud computing environment. Both the above techniques aimed at enhancing the performance and optimizing the resource provisioning in the cloud, but somehow neglected the cost awareness for resources.

Cloud resources and data centers have been aiming at reducing energy consumption [20]. The technique aims at providing a real time service for a virtual request. Different schemes have been proposed to reduce energy consumptions and enhance the performance through simulations [16, 20]. Dynamic load distribution policies have been proposed by Kien and Jingru [21] that addresses the cost related to electricity and the cooling effects. Load spikes have been handled through different cooling strategies, a comparison have been done between the cost aware and cost unaware policies by addressing the cost saving related to electricity and load migration.

Reliability Profit Assurance (RPA) algorithm [22] was presented to investigate the reliability of resources in distributed computing systems in context of operating costs and scheduling. To increase reliability, RPA algorithm introduced cost aware replication mechanism. This research made a first attempt to evaluate the reliability cost in workflows but made many gaps in the research which are addressed in our system. Resource selection strategy [23] is presented to select a host based on the minimum network delay. The aim of this model is to minimize the time between requesting and retrieving a service from cloud provider by selecting the host that is the closest in the network. And it minimizes the execution time of the tasks. The approach was good but it only focused on the location of the cloud provider, minimizing the distance between the client and provider to decrease the cost whereas our proposed system covers other necessary factors other than location for the best selection of cloud provider that charges the lowest cost.

Resource scheduling and optimization is an emerging paradigm in the field of cloud computing [24, 25]. The scheduling algorithm has been introduced in the hybrid clouds presenting the important aspects when scheduling workflows [24]. The communication channels are accessed on job allocation and the performance of scheduling algorithms has been evaluated by comparing the impact of the available bandwidth. For scheduling purpose, the tasks are divided and grouped into the requested resource and data, and then are prioritized [25]. Resource selection is done using a priority formula over sequential scheduling but it does not provide the best resource provider that provides cost aware resources on user demands. Cost aware allocation of cloud resources have been presented by Markus and Simon [26]. In their work, workload forecasting model has been introduced based on Fourier transformation. The aim of the paper was to achieve efficiency of resource utilization. The work aimed at scheduling and improving the resource utilization but did not focus on the cost awareness of resource selection in inter clouds, which is best presented in my proposed system. HCOC [28] addressed which resources should be requested from the public cloud and merged with the private cloud to provide the best results within a given execution time. The aim was to achieve the desired result at a given execution time. The results were then evaluated through simulations. A delay-constrained optimization framework [27] has been proposed on the basis of cost models, to minimize the cloud overhead and maximize the resource utilization. The total execution time is reduced by selecting the appropriate mapping nodes for the modules that have been assigned priority.

Selection and binding of resources at an optimal cost is discussed [30]. The focus of the paper was to improve the quality of results by achieving efficiency, robustness, and scalability. A prototype implementation was also presented for the resource selection and binding component. Different parameters were analyzed to estimate the global energy needed for the development of a service or system. The types of energy like operating energy and embodied energy were discussed [29]. But the only factor in focus was –energy- whereas my proposed system focuses on -cost- as the main factor for service selection from anywhere in the world.

Hybrid modeling [31] focuses on the complex structure of resource selection systems having the ability to cope up with the enterprise architecture. It focuses on the simulation techniques to decrease the involvement of stakeholders and third parties as compared to other techniques. But the paper only focused on the scheduling of the services ignoring the factor of cost which is deeply covered in my proposed system.

III. Proposed Framework

In my proposed solution, I present a new cost aware resource selection algorithm. The user will give the input based on two approaches provided by the algorithm i.e. Instance based and Quantity based approaches. Based on these inputs, the algorithm will display the best cloud providers that charge the lowest cost for the specified services.

A. Proposed Cost Aware Resource Selection Algorithm

The system mainly takes input from the users based on the factors that influence cost. The first part of this research is to find out those factors and is termed as parameters for which user input is obtained. My system decides the best cloud provider among Amazon EC2, Cloudsigma, Google, GoGrid, and Windows Azure, considering the given user input and displays the optimal costs of every cloud provider. It also displays the lowest cost from the list pointing out the most pertinent cloud provider in terms of cost. The system architecture of the proposed cost aware resource selection system is shown in Fig. 2. The system only focuses on IaaS.

The database designed for the proposed system holds all the detailed information about the cloud providers. It also stores the user requirements in a separate table. The database holds the type of instances of the clouds, the number of cores for every particular instance, RAM (in GB), computational storage (in GB), standard storage, bandwidth in, bandwidth out, location, operating system, contract period and cost for every particular instance and their combinations.

Cloud ecosystem is also integrated with the proposed model. Through cloud ecosystem integration services, cost aware resource selection algorithm can reduce up-front infrastructure capital and maintenance costs while also reducing or keeping their in-house infrastructure footprint or inventory under control.

The users will be enabled to provision Windows and Linux operating systems. The system's consistency will enable them to use the same VMs and management tools on the platform that they use on their premises, thus reducing the costs.

- Virtual machines
- Storage, backup and recovery
- Big compute

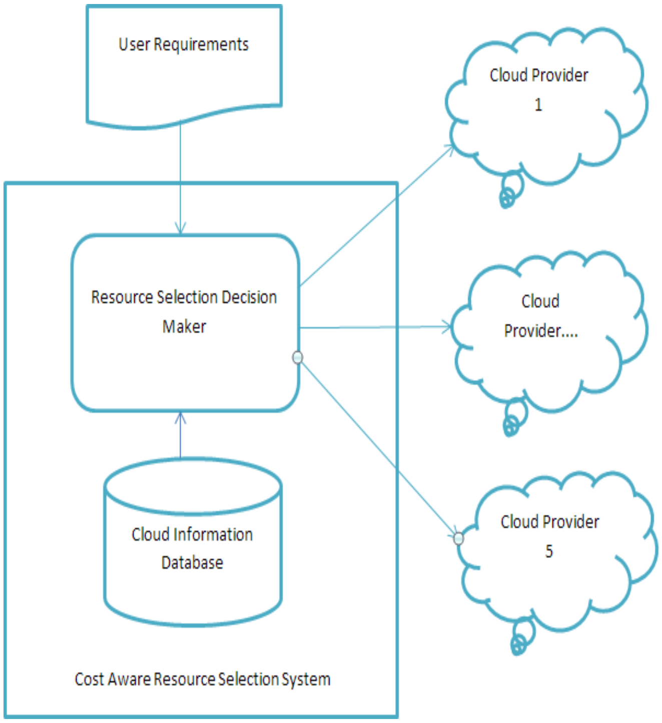

User Requirements
Cloud Provider 1
Resource Selection Decision Maker
Cloud Provider....
Cloud Information Database
Cloud Provider 5
Cost Aware Resource Selection System

Fig. 2. System Architecture of the Proposed Cost Aware Resource Selection System.

B. Detailed Design

Cost aware resource selection system takes the user requirements based on the identified parameters, analyses them, and provides the most pertinent cloud provider in terms of cost. The components of the system are designed in MVC (Model-View-Controller) model to reduce dependencies among them. The MVC architecture of the system is shown in Fig. 3.

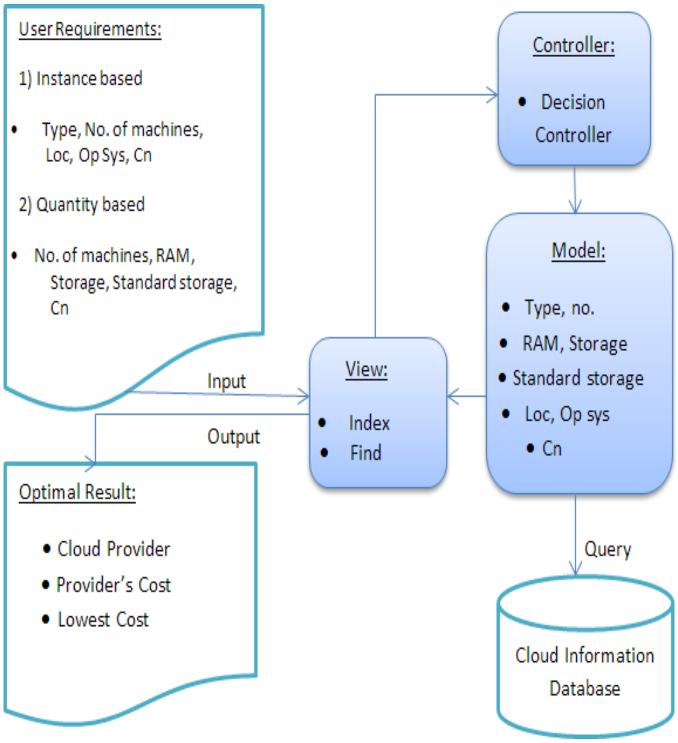


Fig. 3. Cost Aware Resource Selection System Architecture for Model-View- Controller.

TABLE I. FACTORS OF USER REQUIREMENTS

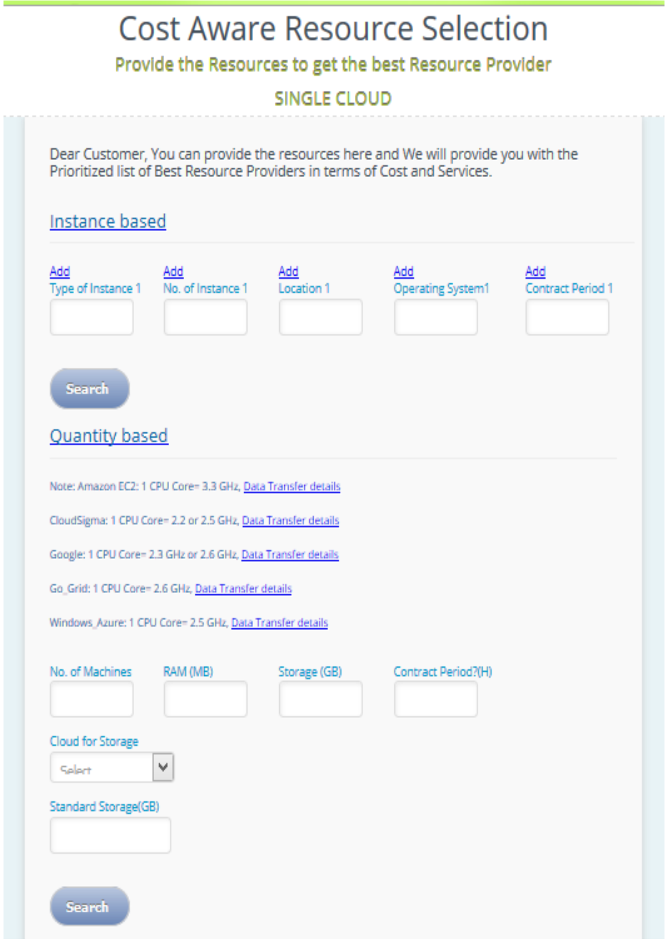
Field	Description	Example
Resource (CPU, RAM, Storage)	Quantity user requires	CPU= 2, RAM= 0.5, Storage= 20
Type of Instances	Every cloud has their particular Instances	t2microinstance
No. of machines	Number of cores user requires	30
Location	User can select US_east or US_west	US_east
Operating System	User can select Linux or windows	Linux
Contract period	It can be hourly, monthly, or yearly	5 hours

IV. IMPLEMENTATION ANDEXPERIMENTATION

A. *Implemenation*

Cost Aware Resource Selection system is implemented using Java scripting and PHP. I have implemented this prototype as a proof of the proposed framework. Five cloud providers have been considered for this research work, from the list of IaaS providers, that is Amazon EC2, Cloudsigma, Google, GoGrid, and Microsoft Windows Azure. A web interface is designed for the user to take the requirements as Input. These web pages have been designed for Single cloud user requirements and Mutli-cloud user requirements.

Fig. 4 shows the Single Cloud web interface for the user to enter requirements. This web page has 2 options for the user i.e. Instance based, and Quantity based. If the User selects the Instance based approach, the user is asked to enter the requirements based on Instance type, number of instances, location, operating system, and contract period. The input is taken and is matched with the records of database under every particular cloud through a query. The cost of the record that matches the user requirements is retrieved from the database and then a list is displayed to the user with all the providers and their respected costs. Then a comparison is made between the costs of five cloud providers. The cost which is lowest is displayed to the user.



Cost Aware Resource Selection

Provide the Resources to get the best Resource Provider

SINGLE CLOUD

Dear Customer, You can provide the resources here and We will provide you with the Prioritized list of Best Resource Providers in terms of Cost and Services.

Instance based

Quantity based

Note: Amazon EC2: 1 CPU Core= 3.3 GHz, Data Transfer details

CloudSigma: 1 CPU Core= 2.2 or 2.5 GHz, Data Transfer details

Google: 1 CPU Core= 2.3 GHz or 2.6 GHz, Data Transfer details

Go_Grid: 1 CPU Core= 2.6 GHz, Data Transfer details

Windows_Azure: 1 CPU Core= 2.5 GHz, Data Transfer details

Fig. 4. Single cloud web interface

In the Quantity based approach section, the user can give the input based on the parameters of number of cores, RAM, Computational storage, contract period. The user can give the standard storage based on the cloud he/she selects. This input is taken and the logic is performed at the run time. In the single cloud, the user can enter the standard storage for a single cloud. The logic is given below:

Amazon Ec2: Storage =
“n” GB Cost per GB= $ 0.03
So, Storage cost= $0.03*n
Provider’s Cost= Amazoncost + storage cost
Here Amazoncost is the cost that is retrieved from the database. Similarly, User selects a standard storage of Cloudsigma, Google, GoGrid, and Windows Azure.
Cloudsigma:
Storage =n Cost per
GB= $ 0.13
So, Storage cost= $0.13*n
Cost= Sigmacost + storage
cost **Google:** Storage =n
Cost per GB= $ 0.026
So, Storage cost= $0.026*n
Cost= Googlecost + storage
cost **GoGrid:** Storage =n
Cost per GB= $ 0.15
So, Storage cost=
$0.15*n Cost= Gridcost
+ storage cost **Windows**
Azure: Storage =n Cost
per GB= $ 0.024
So, Storage cost= $0.024*n
Cost= Windowscost + storage cost

After taking the input from the user, the list with optimal results is displayed to the user. After displaying a list with five cloud providers and their respective costs, a comparison is done between these costs and the results are shown in Table IV.

B. *Algorithm Logic and Flow*

Cost aware resource selection algorithm has been split into two basic algorithms. One is through using the Instance based approach while the other is through using Quantity based approach. Using instance-based approach; the user provides the input in the form of type of instance, number of instances, location, operating system, and contract period. The variable “n” defines the number of additions to an instance. The maximum limit has been defined as 10.

Optimal cost is defined as the cost of the output from the cloud provider’s database based on the factors like RAM (GB) CPU cores, computational storage, location, and the communication cost, computation cost, additional storage cost, cost of network bandwidth in and cost of network bandwidth out. Mathematically we can formulate as follows: Let Database cost= DBC, Communication cost= ComC, Computation cost= CtC, additional storage cost= ASC, Network bandwidth In Cost= NBIC, Network Bandwidth Out Cost= NBOC.

So, Optimal Cost= DBC+ComC+Ctc+ASC+NBIC+NBOC

Algorithm 1: Cost aware resource selection algorithm using Instance based approach

Input: maxIns: the number of maximum iterations for instances;

Ti: type of instance;

Ni: number of instances; Li: location;
Osi: operating system; Cni: contract period;
Output: C1, C2, C3, C4, C5: Costs of Amazon EC2, Cloudsigma, Google, GoGrid, Windows Azure
Lc: Lowest optimal cost

```
1. Lc=NULL; length=0; n=length+1;
2. while (n<10) do
// Td, Ld, Osd are the saved values of particular cloud providers
3. if Ti=Td AND Li=Ld AND Osi=Osd then
3.1. set C1= saved cost*N1*Cn1;
3.2. set C2= saved cost*N2*Cn2;
3.3. set C3= saved cost*N3*Cn3;
3.4. set C4= saved cost*N4*Cn4;
3.5. set C5= saved cost*N5*Cn5;
4. else C1&C2&C3&C4&C5 = NULL;
5. end while
6. if C1≠0 then
6.1. if C1<C2 AND C1<C3 AND C1<C4 AND C1<C5 then
6.2. UPDATE Lc=C1;
6.3. end if;
7. if C2≠0 then
7.1. if C2<C1 AND C2<C3 AND C2<C4 AND C2<C5 then
7.2. UPDATE Lc=C2;
7.3. end if;
8. if C3≠0 then
8.1. if C3<C1 AND C3<C2 AND C3<C4 AND C3<C5 then
8.2. UPDATE Lc=C3;
8.3. end if;
9. if C4≠0 then
9.1. if C4<C1 AND C4<C2 AND C4<C3 AND C4<C5 then
9.2. UPDATE Lc=C4;
9.3. end if;
10. if C5≠0 then
10.1. if C5<C1 AND C5<C2 AND C5<C3 AND C5<C4 then
10.2. UPDATE Lc=C5;
10.3. end if;
11. end if;
12. return Lc;
```

Now, the second algorithm states the Quantity based approach. The user gives the input inform of quantities like number of machines, the amount of RAM, computational storage, the standard storage which is unique for every particular cloud and for that the user chooses the cloud too, and finally the contract period.

Algorithm 2: Cost aware resource selection algorithm using Quantity based approach

Input: Ni: number of machines; Ri: RAM;
Si: computational storage; SSi: standard storage;
Ci: cloud provider Cni: contract period;
Output: C1, C2, C3, C4, C5: Costs of Amazon EC2, Cloudsigma, Google, GoGrid, Windows Azure
Lc: Lowest optimal cost

```
1. Lc=NULL; C1=0; C2=0; C3=0; C4=0; C5=0;
2. while input fields are not empty do
// Td, Ld, Osd are the saved values of particular cloud providers
3. if Ri=Rd AND Si=Sd then
3.1. if Ci= Amazon EC2 then;
3.1.1. set A=0.03*SSi;
3.1.2. UPDATE C1= saved cost*Cni*Ni + A;
3.2. if Ci= Cloudsigma then;
3.2.1. set C2=0.13*SSi;
3.2.2. UPDATE C2= saved cost*Cni*Ni + B;
3.3. if Ci= Google then;
3.3.1. set C=0.135*SSi;
3.3.2. UPDATE C3= saved cost*Cni*Ni + C;
3.4. if Ci= GoGrid then;
3.4.1. set D=0.15*SSi;
3.4.2. UPDATE C4= saved cost*Cni*Ni + D;
3.5. if Ci= Windows Azure then;
3.5.1. set E=0.024*SSi;
3.5.2. UPDATE C5= saved cost*Cni*Ni + E;
4. else C1&C2&C3&C4&C5 = NULL;
5. end while
6. if C1≠0 then
6.1. if C1<C2 AND C1<C3 AND C1<C4 AND C1<C5 then
6.2. UPDATE Lc=C1;
6.3. end if;
7. if C2≠0 then
7.1. if C2<C1 AND C2<C3 AND C2<C4 AND C2<C5 then
7.2. UPDATE Lc=C2;
7.3. end if;
8. if C3≠0 then
8.1. if C3<C2 AND C3<C1 AND C3<C4 AND C3<C5 then
8.2. UPDATE Lc=C3;
8.3. end if;
9. if C4≠0 then
9.1. if C4<C2 AND C4<C3 AND C4<C1 AND C4<C5 then
9.2. UPDATE Lc=C4;
9.3. end if;
10. if C5≠0 then
10.1. if C5<C2 AND C5<C3 AND C5<C4 AND C5<C1 then
10.2. UPDATE Lc=C5;
10.3. end if;
11. end if;
12. return Lc;
```

C. Cloud Information Database

Cloud Information database includes the details about the five clouds i.e. Amazon EC2, Cloudsigma, Google, GoGrid, and Windows Azure. It also includes a table of User requirements that saves the input taken by the user. The information database holds the records of the cloud providers and their details including type of instances, CPU, RAM, storage, standard storage, bandwidth in, bandwidth out, location, operating system, contract period and the pricing details.

I had discussed in the related work section that different cloud providers have their pricing strategies and few of the providers have their own pricing policies which may be totally vary from the rest ones. In Cost aware resource selection system, all the information should be stored in our cloud information database consistently so that the system can display the optimum result without taking help from any otherprograms. Primary keys have been assigned to every particular provider’s table and have explicit details about resources and pricing.

Different combinations have been made in the database for the ease of user to select any kind of combination and find the optimum result.

Following few combinations has been shown using Linear Programming Model (LPM) for better understanding:

I= {Ia1, Ia2 , ..., Ian} to {Iw1, Iw2 , ..., Iwn}
(Amazon=a...Windows_azure=w)
CP= {CPa1, CPa2 , ..., CPan} to {CPw1, CPw2 , ..., CPwn}
(Amazon=a...Windows_azure
=w)
R= {Ra1, Ra2 , ..., Ran} to {Rw1, Rw2 , ..., Rwn}
(Amazon=a...Windows_azure=w)
S= {Sa1, Sa2 , ..., San} to {Sw1, Sw2 , ..., Swn}
(Amazon=a...Windows_azure
=w) OS = {OS1, OS2}
Loc = {Loc1, Loc2}
Cn = {Cn1, Cn2, ..., Cnn}

Where n>=1, I is the set of n instances, CP is the set of n CPU cores, OS is the set of 2 operating systems, R is the set of n RAMs, Loc is the set of 2 locations, S is the set of n Storage, and Cn is the set of n Contract period.

We have introduced few notations to be used in the paper. They are:

I is the instance for every particular cloud, CP is the CPU cores, OS is the type of operating systems, R is the RAM, Loc is the type of location, S is the Storage size, Cn is the Contract period. Let the cloud providers be represented as Amazon_Ec2= a, Cloudsigma=c, Google=g, Go_grid=gg, and Windows_Azure=w.

The database is designed with multiple combinations. The design of the database consists of the fields shown in Table III. The combinations in the database are represented using the linear programming terminologies shown in the following Table II:

The Cost aware resource selection system itself is a Linear Programming (LP) problem. The objective function and the set of constraints are as follows:

Minimize
$C_{min} = \Sigma\ i{=}1n\ InCPnRnSnLocnOSn$ (1)

Subject to
$n>0;$ (2)
$Locn \in \{1,2\}$
$OSn \in \{1,2\}$ (3)

The objective function (1) is the computational cost, which we wish to minimize. With reference to constraint (2), the value of Locn shows whether it can be Loc1 which represents US_east, or Loc2 which represents US_west. Constraint (3) shows that OSn can either be OS1 which represents Linux, or OS2 which represents Windows.

TABLE II. REPRESENTATION OF COMBINATIONS OF DATABASE

Main variables/ Cloud providers	**Amazon a**	**Cloud sigma c**	**Google g**	**Go-Grid gg**	**Windo ws Azure w**
I1	Ia1	Ic1	Ig1	Igg1	Iw1
I2	Ia2	Ic2	Ig2	Igg2	Iw2
:	:	:	:	:	:
In	Ian	Icn	Ign	Iggn	Iwn
CP1	CPa1	CPc1	CPg1	CPgg1	CPw1
CP2	CPa2	CPc2	CPg2	CPgg2	CPw2
:	:	:	:	:	:
CPn	CPan	CPcn	CPgn	CPggn	CPwn
R1	Ra1	Rc1	Rg1	Rgg1	Rw1
R2	Ra2	Rc2	Rg2	Rgg2	Rw2
:	:	:	:	:	:
Rn	Ran	Rcn	Rgn	Rggn	Rwn
S1	Sa1	Sc1	Sg1	Sgg1	Sw1
S2	Sa2	Sc2	Sg2	Sgg2	Sw2
:	:	:	:	:	:
Sn	San	Scn	Sgn	Sggn	Swn
Cn1	Cna1	Cnc1	Cng1	Cngg1	Cnw1
Cn2	Cna2	Cnc2	Cng2	Cngg2	Cnw2
:	:	:	:	:	:
Cnn	Cnan	Cncn	Cngn	Cnggn	Cnwn

TABLE III. FIELDS OF CLOUD INFORMATION DATABASE

Field	**Description**	**Example**
Row_id	Primary key to uniquely identify every provider	Row_id=1
Resource (CPU, RAM, Storage)	Resource set of every cloud	CPU= 4, RAM=0.75, Storage=30
Type of Instances	Every cloud has their particular Instances	minimum
No. of machines	Number of cores user requires	50
Location	2 locations are saved i.e. US_east or US_west	US_west
Operating System	2 choices are saved i.e. Linux or windows	Linux
Contract period	It can be hourly, monthly, or yearly	5 hours

D. Results

In Experiment 1, User Selects “Instance based” approach and gives input. In Experiment 2, User selects Quantity based approach and gives Input. The results for both are shown in Table IV. In Experiment 3, User Selects “Instance based” approach and gives input. The results are same as Experiment 1 because this search is based on instance type, and every cloud has their unique instances. No instance of one cloud matches with the instance of other cloud. Hence, in the multi cloud scenario, the results for “Instance based approach” will be same as that of single cloud. In Experiment 4, user selects Quantity based approach and gives Input. Scenarios are taken for Cloudsigma: User gives the input but selects the standard storage of any other cloud other than CloudSigma itself, like standard storage of Amazon EC2, Google, GoGrid, or Windows Azure, to validate that the standard storage can be chosen from any other cloud now.

TABLE IV. EXPERIMENTATION RESULTS

Type of cloud	Approach	Input resources	Optimal cloud	Optimal cost ($)
Single	Instance based	Type=t2microinstance, No.= 1, Loc= us_east ,OS= linux, Contract period= 2hrs	Amazon EC2	0.026
Single	Quantity based	No. of machines=1 , RAM (MB)= 0.5, Storage (GB)= 0.5, Contract period= 1, Cloud for storage= Cloudsigma, Standard storage (GB)= 2	Gogrid	0.29
Multi	Instance based	Type=t2microinstance, No.= 1, Loc= us_east ,OS= linux, Contract period= 2hrs	Amazon EC2	0.026
Multi	Quantity based	No. of machines=1 , RAM (MB)= 0.5, Storage (GB)= 0.5, Contract period= 1, Cloud for storage= Amazon EC2, Standard storage (GB)= 2	Gogrid	0.09

V. CONCLUSION

This paper aims on finding the best resource provider in terms of lowest cost. The system administrators usually spent a lot of time to find the cloud providers with the type of resources they provide and go through the whole pricing policy details. Through cost aware resource selection algorithm, the solution is provided to this particular problem. I reviewed the old approaches where the focus is not on a single factor i.e. cost but also on multiple other factors like efficiency, performance, power, reliability etc. which somehow made the researchers lose the focus on cost. My proposed framework mainly focuses on providing the optimal cost for Single cloud and Multi clouds. I have described the detailed architecture of my cost aware resource selection system and have implemented the prototype. I have deduced some common factors based on our literature review that affected cost in any possible way i.e. Instance type, RAM, CPU cores, computational storage, location, operating system, contract period, number of machines and standard storage. Based on these factors an interface is designed using PHP and java scripting. Input is taken from the user and is matched with that of our cloud information database. Cloud information database is designed in phpMyAdmin to maintain the details of every particular cloud and their pricing policies. The clouds taken for our research are Amazon EC2, Cloudsigma, Google, GoGrid and Microsoft Windows Azure. User requirements are matched with that of the database records and a list is displayed to the user with the costs of all the clouds. Then comparison logic is run to find the optimal and lowest cost among them. This is done under Instance- based approach whereas in Quantity-based approach the results are made through runtime logic. Experimentation and evaluation have proved the validity of the system.

For future work, the parameters that I took for my prototype can be enhanced and increased. Many other locations can be added and more operating systems can be supported in my research. There are many issues that remain open. Something that can search the web interface and updates the cloud information database can be further investigated.

REFERENCES

[1] Buyya, Rajkumar, et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems 25.6 (2009): 599-616.

[2] Vaquero, Luis M., et al. "A break in the clouds: towards a cloud definition." ACM SIGCOMM Computer Communication Review 39.1 (2008): 50-55

[3] Mell, Grance. "Peter Mell, Timothy Grance: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology." Gaithersburg, Maryland (2011)

[4] Motta, Gianmario, Nicola Sfondrini, and Daniele Sacco. "Cloud computing: An architectural and technological overview." 2012 International Joint Conference on Service Sciences. IEEE, 2012.

[5] Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58

[6] Zhang, Miranda, et al. "Investigating decision support techniques for automating cloud service selection." arXiv preprint arXiv:1210.3077 (2012).

[7] M. Zhang, R. Ranjan, S. Nepal, M. Menzel and A. Haller, "A declarative recommender system for cloud infrastructure services selection," in Proceedings of the 9th international conference on Economics of Grids, Clouds, Systems, and Services, Berlin, Germany, Springer, 2012.

[8] Sharma, Upendra, et al. "A cost-aware elasticity provisioning system for the cloud." 2011 31st International Conference on Distributed Computing Systems. IEEE, 2011.

[9] Liu, Jiangchuan, et al. "A novel framework for QoS-aware resource discovery in mobile ad hoc networks." Communications, 2002. ICC 2002. IEEE International Conference on. Vol. 2. IEEE, 2002.

[10] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications 1.1 (2010): 7-18.

[11] Zeng, Wenying, Yuelong Zhao, and Junwei Zeng. "Cloud service and service selection algorithm research." Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation. ACM, 2009.

[12] Guo, Tian, et al. "Cost-aware cloud bursting for enterprise applications." ACM Transactions on Internet Technology (TOIT) 13.3 (2014): 10.

[13] Wang, Lijuan, Jun Shen, and Ghassan Beydoun. "Enhanced ant colony algorithm for cost-aware data-intensive service provision." 2013 IEEE Ninth World Congress on Services. IEEE, 2013.

[14] Grozev, Nikolay, and Rajkumar Buyya. "Inter-Cloud architectures and application brokering: taxonomy and survey." Software: Practice and Experience 44.3 (2014): 369-390.

[15] Sharma, Upendra, et al. "Kingfisher: Cost-aware elasticity in the cloud." INFOCOM, 2011 Proceedings IEEE. IEEE, 2011.

[16] Jung, Gueyoung, et al. "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures." 2010 International Conference on Distributed Computing Systems. IEEE, 2010.

[17] Alicherry, Mansoor, and T. V. Lakshman. "Network aware resource allocation in distributed clouds." Infocom, 2012 proceedings IEEE. IEEE, 2012.

[18] Li, Jiayin, et al. "Online optimization for scheduling preemptable tasks on IaaS cloud systems." Journal of Parallel and Distributed Computing 72.5 (2012): 666-677.

[19] Ferrer, Ana Juan, et al. "OPTIMIS: A holistic approach to cloud service provisioning." Future Generation Computer Systems 28.1 (2012): 66-77.

[20] Kim, Kyong Hoon, Anton Beloglazov, and Rajkumar Buyya. "Power-aware provisioning of cloud resources for real-time services." Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science. ACM, 2009.

[21] Le, Kien, et al. "Reducing electricity cost through virtual machine placement in high performance computing clouds." Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ACM, 2011.

[22] Lee, Young Choon, Albert Y. Zomaya, and Mazin Yousif. "Reliable workflow execution in distributed systems for cost efficiency." Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on. IEEE, 2010.

[23] Lakhani, Jignesh, and Padam Kumar. "Resource selection strategy based on propagation delay in Cloud." Communication Systems and Network Technologies (CSNT), 2012 International Conference on. IEEE, 2012.

[24] Bittencourt, Luiz F., Edmundo RM Madeira, and Nelson LS Da Fonseca. "Scheduling in hybrid clouds." IEEE Communications Magazine 50.9 (2012).

[25] Dr Ajay jangra, Tushar Saini,“ Scheduling Optimization in Cloud Computing”, IEEE, Journal paper, Volume 3, Issue 4, April 2013

[26] Hedwig, Markus, Simon Malkowski, and Dirk Neumann. "Towards Autonomic Cost-Aware Allocation of Cloud Resources." ICIS. 2010.

[27] Zhu, Mengxia, Qishi Wu, and Yang Zhao. "A cost-effective scheduling algorithm for scientific workflows in clouds." Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International. IEEE, 2012.

[28] Bittencourt, Luiz Fernando, and Edmundo Roberto Mauro Madeira. "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds." Journal of Internet Services and Applications 2.3 (2011): 207-227.

[29] Dixit, Manish Kumar, et al. "Identification of parameters for embodied energy measurement: A literature review." Energy and Buildings 42.8 (2010): 1238-1247.

[30] Kee, Yang-Suk, et al. "Efficient resource description and high-quality selection for virtual grids." Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on. Vol. 1. IEEE, 2005.

[31] Jahangirian, Mohsen, et al. "Simulation in manufacturing and business: A review." European Journal of Operational Research 203.1 (2010): 1-13.