# Probabilistic Model for Resource Demand Prediction in Cloud

**Niraja Jain[a], Dr B Raghu[b], Dr V Khanaa[c]**

[a]Research Scholar, Computer Engineering Department, Bharath University, Chennai, India
[b] Principal, SVS Group of Institutes, Warangal, India
[c]Dean, Bharath University, Chennai, India
Email: [a]nijajain@gmail.com, [b]balrajraghu@gmail.com, [c] khanaakrishna@gmail.com

**Article History**: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

**Abstract:** Dynamic cloud infrastructure provisioning is possible with the virtualization technology. Cost, agility and time to market are the key elements of the cloud services. Virtualization is the software layer responsible for interaction with multiple servers, bringing entire IT resources together and provide standardized Virtual compute centers that drives the entire infrastructure. The increased pooling of shared resources helps in improving self-provisioning and automation of service delivery. Probabilistic model proposed in this article is based on the hypothesis that the accurate resource demand predictions can benefit in improving the virtualization layer efficiency. The probabilistic method, uses the laws of combinatorics. The probability space gives an idea about both the partial certainty and randomness of the variable. The method is popular in theoretical computer science. The probabilistic models provide the predictions considering the randomness of the variables. In the cloud environment there are multiple factors dynamically affecting the resource demand needs. The resource demand has a certain degree of certainty but the randomness of requirements. This further leads to decrease in risk related to leveraging cloud services. It accelerates development and implementation of cloud services that overall improves the services pertaining to SLA

**Keywords:** Cloud Computing, Virtualization, Resource organization, Resource optimization, Probabilistic model, Machine learning

## 1. Introduction

Cloud computing is the environment that enhances the user experience by providing a bouquet of services over the virtual infrastructure. Deploying the applications onto cloud undergoes multiple stages of verification and validation at the service provider and service user end. Though it is a tedious and complex job, the basic components involves:

Provisioning:
Installation & Configuration:
Fault management:

Many companies wish to deliver their software product to the end user over cloud as a service. Cloud computing supports the service provisioning on pay-as-you-go model [1]. The fundamental features of Cloud computing are elasticity and scalability that can be exploited to leverage the services by creating multiple virtual instances as required by the end user[1,2]. The cloud computing architectures commonly available in following forms:

Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service(IaaS)

SaaS is the most commonly available and used model of cloud computing wherein the software is made available off-premises with the scalability and elasticity to cater to the growing/shrinking demand of the end user. [3]. While catering to this need the resources under utilization are only charged and this facility makes it convenient and popular over the traditional systems [1].

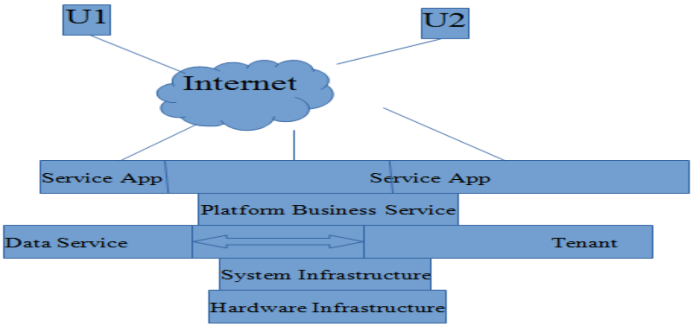


**Fig. 1** Software as a Service SaaS Architecture

## 2. Motivation

The Cloud computing has many advantages and different service providers like Google, Amazon, Microsoft, VMWare, IBM and many more in the market are offering the services at different cost and functionality. All of them do have the major concern regards the resource utilization during the low and peak load situations. Either the resources are under-utilized or over-utilized in absence of the dynamic elasticity and scalability [6,7]. The most natural strategy for resource allocation is pre-booking them based on the past data analysis for assumed workloads. Most of the cloud service providers do charge for the instances being boked even though the system was idle due to non-availability of the load as predicted earlier. This in turn hampers the essentail basics of cloud computing that is pay-as-you-go model.

Armbrust et al. [1] provide a calculation of this problem stating that around 1.7 times more than the needed resources are engaged.

Over-utilization is another aspect where customer is compelled to stop utilizing the services due to saturation resulting in revenue loss[1]. Current cloud platforms are unable to provide the real elastic model with full potential.

SaaS is expected to provide the facility of releasing and acquiring the resources as and when needed guranteeing the system efficiancy and cost effective payment model[9]. The traditional method of resource allocation model charges the user for global usage rather than taking into account the actual utilization by the end user. This leads to the need to create a true elastic architecture to charge SaaS providers the actual resource usage [6].

To achieve cost-effective SaaS scalability, a level of automation is necessary. This intelligent environment can be achieved through the probabilistic model of machine learning that is aware of the user demand and actual usage[10]. With the use of cloud computing approaches such as on-demand resource allocation through SOAP interfaces, it is possible to efficiently create virtualized resources for SaaS applications which allows to allocate and charge only consumed resources in a tenant-based environment.

This article discusses in detail the under and over-provisioning of virtualized resources (i.e. memory used and CPU utilization. Herein propose a machine learning model based on probabilistic methods to predict the resource demand which will be helpful in VM instance allocation and load balancing to make the intelligent cost effective cloud computing environment.

## 3. Objective

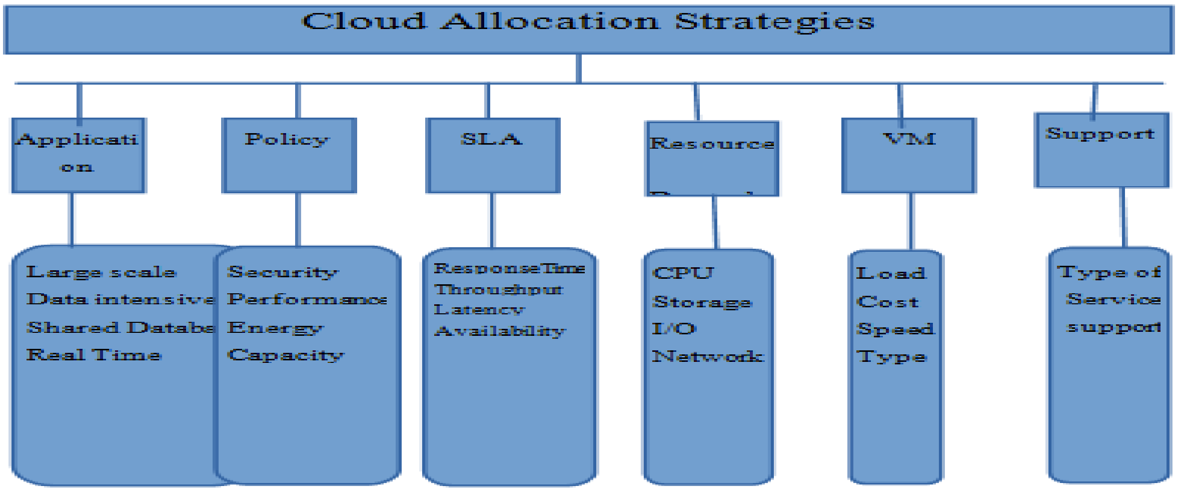


**Fig.2** Cloud Allocation Strategies

Cloud allocation strategies studied from the literature are depicted in fig.2. The basics of the VM is the load to be served and the corresponding cost involved for the type of load. There are multiple strategies for resource allocation which includes the appropriate resource selection. The most common resource selection algorithms such as GlOSS, CORI, ReDDE, Geometric Average and other classification-based method focuses on the user information to check the relevance of the available sources.

Imagine that a client has developed a web based search service that is available to the world for use through WEB2. Cloud computing will enable the developer to host this service remotely and can deal with the sacle variability efficiently. As the business grows or shrinks, developer can acquire or release the resources easily and relatively inexpensively. On the other hand, implementation and maintenace of the data services that are scalable and adaptable to such dynamic conditions becomes a challenge. Especially when the data services are the compositions of the other possibly third party services (eg., Google search or Yahoo Image search), these serivces becomes the data processing graphs that use the third party services as building blocks and invoke them during their execution. Running these data services under different QoS (Quality of Service) constraints as per the clients requirements further makes the system complex.

To decide upon which third party services to be used in processing correctly, making the optimal use of the resources available, satisfy all the QoS constraints is quite difficult. To make data services scalable and adaptable to the cloud environment, the dataflow needs to be optimized automatically.

Current algorithms do not model the important relationship information among individual sources. For example, an information source tends to be relevant to a user query if it is similar to another source with high probability of being relevant. This paper proposes a joint probabilistic classification model for resource selection. The model estimates the probability of relevance of information sources in a joint manner by considering both the evidence of individual sources and their relationship. An extensive set of experiments have been conducted on several datasets to demonstrate the advantage of the proposed model.

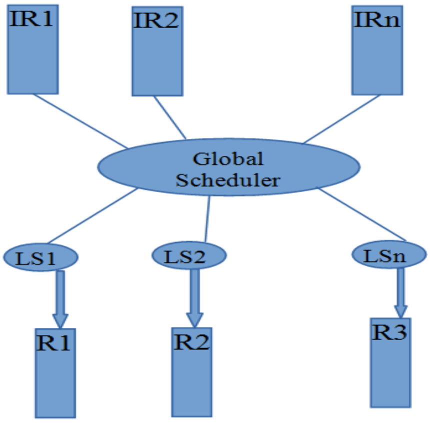
VM management primarily involves two points: efficient and intelligent VM Management and resource mapping. Eucalyptus, provides on demand VM instance deployment. Amazon offers the auto-scaling facility depending on the user demand needs that launches or terminates the EC2 instances. This is being achieved by the commands that run the multiple instances which is sole properitery of Amazon. This is based on the resource utilization not considering the VM instances.

## 4. Methodology

Every future event has the degree of certainty attached to it. This is termed as Probability. More random the event, more random this probabaility function will be. Deterministic functions are the exact opposites of the probabilistic fucntions that do not add event randomness. Probabilistic method or model is based on the theory of probability. Bell Curve is the normal distribution curve which is the building block of the Probabilistic model.

The cloud resources are primarily allocated based on the pre-decided policy agreed upon by the the vendor and the tenant. This article proposes a model that helps in identifying the future demands for partiicular resources. This in turn will help improve and satisfy the QoS.

As shown in the following figure the, the incoming request is sent to the global scheduler. The local scheduler is communicated about the resource requirement. Based on the avaiolability the demanded resources are being made available. Here the primary assumption is that the resource demand is communicate dto the vendor well in advance to avoid on any further delay of processing the requests.



## 5. Contribution

Experimental setup consists of private cloud infrastructure with multiple nodes attached as cloud tenants to utilize the services. The probabalistic model is mathematically described as follows:

D{d1, d2, … dn}: Datanodes

Q{q1, q2, … qn}: Queries by user

N{n1, n2, …, nn}: Computing nodes

R{0/1}: Relevance

P: Total number of queries

For optimized storage resource utilization is desirable to predict the frequency of probable queries arising from different users such that:

f(p(d,q/r=1)

It's a function that predicts the probability of a relevant query by an end user arrives at the corresponding datanode where the primary data or replica resides. The probability ranges from 0 to 1 for exact match or mismatch on the relevance ranking.

The objective is to minimize the turnaround time of the query over the computing nodes by assigning the relevant datanodes to the corresponding query. We present four strategies to analyse whether a new query request can be accepted or not based on the QoS requirements and resource capabilities.

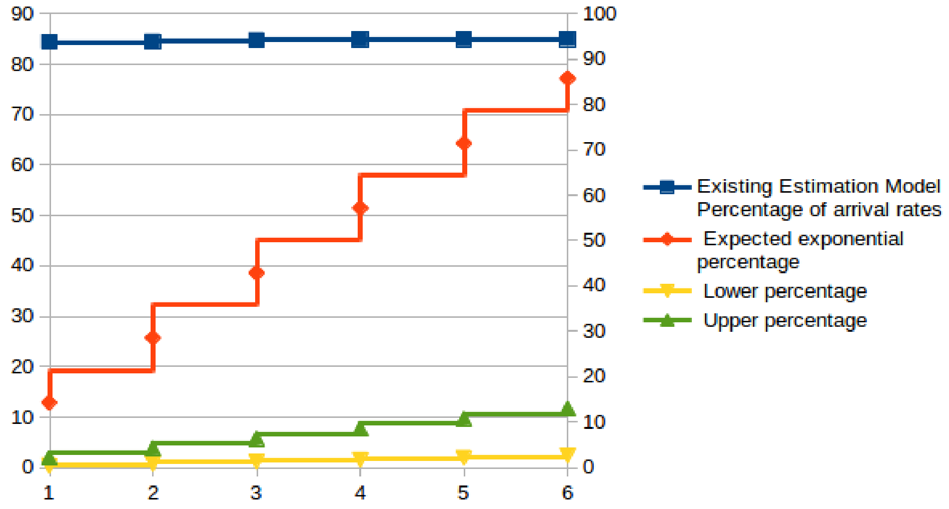
a) initiate new VM,

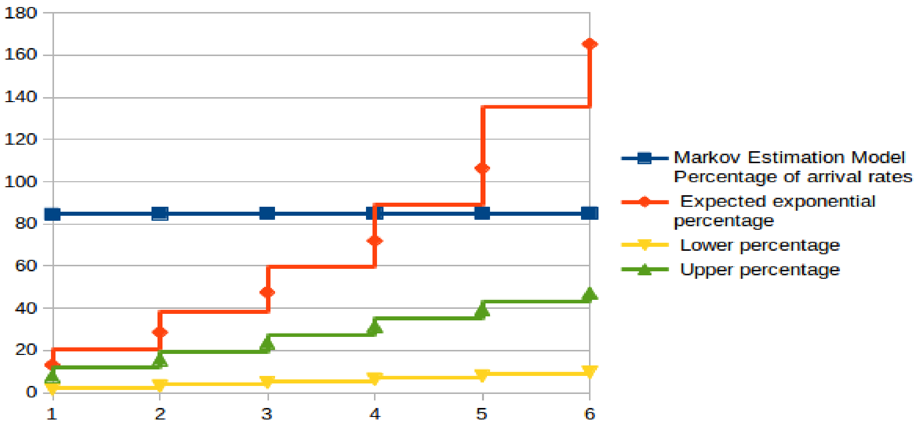
b) queue up the new user request at the end of scheduling queue of a VM,

c) insert (prioritize) the new user request at the proper position before the accepted user requests and,

d) delay the new user request to wait all accepted users to finish.

## 6. Result & Discussions





The dynamic resource allocation is  difficult task because of the everchanging resource demand. This article presents effective computer assisted technique for prediction of resource demand. The probabalistic model understands the over and under utilization curves properly.   The proposed model is evaluated using different parameters as explained below.

– True Negative (TN)

– True Positive (TP)

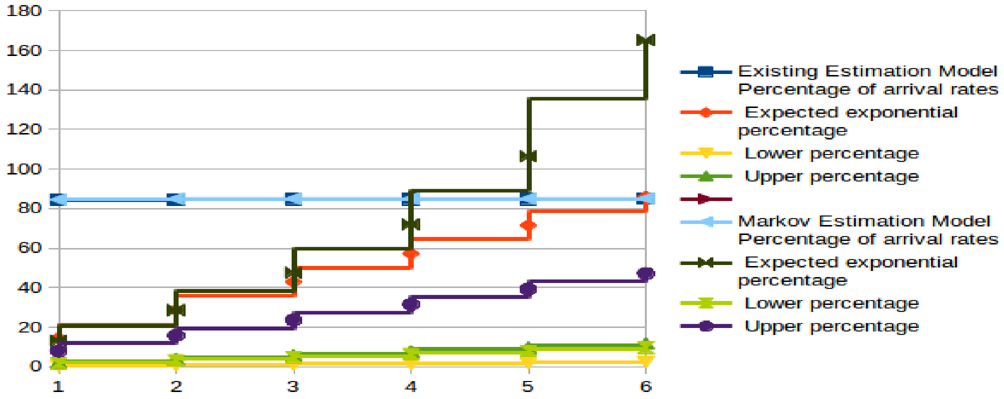
- False Negative (FN)

– False Positive (FP)

## 7. Conclusions

This article is an attempt to discuss the dynamic resource allocation problem and to propose a solution by probabilistic model to predict the resource demand in advance, The resources like CPU, memory and storage available on Virtual machines are either under or over-utilized in absence of the exact identification of the resource demand. The model attempts to predict the demand fairly accurately increasing the cloud efficiency.

While experimenting with this probabalistic model it's observed that the model gives fairly accurate resource demand predictions. In future the model can be extensively tested for the heterogeneous cloud environment for it's prediction accuracy.



## References

1. M. Armbrust, et al. Above the clouds: a Berkeley view of cloud computing, electrical engineering and computer sciences, Technical Report No. UCB/EECS- 2009-28, University of California at Berkeley, February 2009
2. R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg, I. Brandic, 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.
3. Ankita Jain, Arun Kumar Yadav, Brijesh Kumar Chaurasia, “A Proactive Approach for Resource Provisioning in Cloud Computing ”, International Journal of Recent Technology and Engineering (IJRTE) , ISSN: 2277-3878, Volume-7, Issue-5S3, February 2019
4. Karlin, S., Taylor, H.M.: A First Course in Stochastic Processes, 2nd edn, pp. 221–228. Academic Press, Cambridge (1975)
5. Hitoshi Motsumoto, Yutaka Ezaki, “Dynamic resource management in cloud environment”, FUJITSU Sci & Tech J, Vol 47, No 3, pg 270-276, July2011.
6. R. Agrawal et al., “ The claremont report on database research”, ACM SIGMOD Record, 37(3):9–19, 2008
7. Daniel J. Abadi, “Data Management in the Cloud: Limitations and Opportunities”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2009
8. Barnes, A.K., Balda, J.C., Escobar-Mejía, A.: A Semi-Markov model for control of energy storage in utility grids and microgrids with PV generation. IEEE Trans. Sustain. Energy 6(2), 546–556. ISSN 1949-3037 (2015)
9. M. Aazam and E.-N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT,” in Proceedings of the IEEE 29th International Conference on Advanced Information Networking and Applications (AINA ’15), vol. 51, no. 5, pp. 687–694, IEEE Computer Society, March 2015