International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

# Resource Allocation in Multi-Cloud Based on Usage Logs

**Lijin P**[*1]
[*1]Department of Computer Science, Pondicherry University,Pondicherry, Puducherry, India


## ABSTRACT

Cloud computing has become an emerging technology that offers a fast and on-demand way to access the services to the users. Cloud can make it possible to access application and data from anywhere. Companies and users can rent the services of a cloud on payment basis so that the cost can be reduced significantly. The most commonly recurring problem in resource allocation is related to optimizing the resource being allocated and to satisfy user requirements. In the resource allocation process, it is imported to fulfill the requirement of the user so that the provider profit is increased and the cost is reduced to the user. This paper proposes a method to predict the requirement of the user and proposed a game theory based algorithm to allocate resources in the multi-cloud.


**Keywords:** cloud providers, multi-cloud, resource allocation

## I. INTRODUCTION

Cloud computing provides various services such as infrastructure as a service, platform as service, and software as a service. These services are based on a pay-as-you-use scenario to the cloud users. It has favored transforming the large IT industry, making software service more attractive and easy to consume. From the user's perspective, cloud computing is allows to use and deploy their application from anywhere and anytime based on their demands. To provide these types of services continuously based on customer demand, various technologies such as virtualization, clustering and application server are deployed. Virtualization is used in the cloud computing for virtualizing operating system,storage system and data center. Cloud providers offer vitualized resources over internet as a meadium. EachVM type consists of different quantities of resources (CPU,memory, disk storage, etc.) It is up to cloudproviders to make intelligent decisions on how to allocate the heterogeneous resources of physical servers to those required VMs.Table 1 shows a set server types and price of each one is listed here. If you observe this, in each server type, the price and configuration differs. There are many cloud service providers Cloud computing providers, such as Amazon Web Services , IBM cloud and Google Compute Engine (GCE) offer cloud computing platforms and various services. Users can allocate, execute, and terminate the resources as needed. Apart from Table 1, currently amazon providing four types of pricing on amazon EC2 that are as follows, on demand pricing, spot pricing, reserved pricing, dedicated pricing. Multiple types of servers are available in these categories. User has to select the suitable one pricing according to them.

The Internet is evolving day by day also there is a major increase in the internet related technologies. The Internet offers different services to satisfying various users' needs. Cloud computing is one of the

**Table 1**
A subset of AWS EC2 instance types and pricing (Amazon EC2 Pricing, 2015).

| Type | ECU | Memory (GB) | Price |
|---|---|---|---|
| m3.medium | 3 | 3.75 | $0.07/h |
| m3.large | 6.5 | 7.5 | $0.14/h |
| m3.xlarge | 13 | 15 | $0.28/h |
| m3.2xlarge | 26 | 30 | $0.56/h |
| c3.4xlarge | 55 | 30 | $0.84/h |
| c3.8xlarge | 108 | 60 | $1.68/h |

**Table 2**
A subset of GCE machine types and pricing (Google Compute Engine, 2015).

| Type | Virtual cores | Memory (GB) | Price (dollar/h) |
|---|---|---|---|
| n1-standard-1 | 1 | 3.75 | 0.07 |
| n1-standard-2 | 2 | 7.5 | 0.14 |
| n1-standard-4 | 4 | 15 | 0.28 |
| n1-standard-8 | 8 | 30 | 0.56 |
| n1-standard-16 | 16 | 60 | 1.12 |

major technology that is a perfect solution to various users needs like, database storage, compute power, applications etc. Cloud providers offer these services and pay for what you use. Each provider having different QOS requirements. It is very hard to select the cloud service provider that is suitable for the user. So the resource allocation in multi-cloud is very important. Every user has certain requirements such as cost, storage space etc. There might be chances of selecting a high configured resource for a small application. This will leads to resource wastage. User requirements may change in future also sometimes request for a huge number of resources in future may cause allocation failure or system hang problem. Another problem is that resource over provisioning and under provisioning. Allocating more resources than required is called resource overflow. Allocating fewer resources than required is called resource underflow. In order to mitigate these issues, this paper presents an approach to predict the future requirements and to allocate the resources to the users based on predicted requirements. The goal of this research is improve resource utilization.

Virtualization is the key characteristic of cloud computing. Different types of VMs are created on the available physical servers. The VMs are composed of different quantities of resources having different configurations. It is the task of cloud providers to allocate the resources of physical machines to those required VMs. In cloud computing, resources are allocated to the various users using resource allocation techniques. As the opportunities of cloud computing Majority of customers moving towards cloud computing and they started their app creation in cloud environment to promote their business operation to end client with low cost, availability etc. User's requirements are considered for efficient allocation of cloud resources. There has been many papers [1][2][3]that allocates resources to the end users by considering QOS goals.

In summary, the main contributions of this paper are as follows.

(i) Requirement prediction algorithm to predict the user's future requirements based on webserver log file.

(ii) The resource allocation system

(iii) A game theory based algorithm to improve the resource utilization.

This paper is organized as follows. In Section II, the background and related work are presented. In Section III, we present requirement prediction algorithm, the architecture of the cloud resource management system and the mathematical model of resource allocation. Evaluation of the proposed system and algorithm is analyzed in Section V. Section VI is the conclusion.

## II. RELATED WORK

*a) Log based QoS identification*

In[2] K.S.Guruprakash and S.Siva Sathya has proposed a method for resource allocation and SMI parameter identification in cloud. This paper gives a method to identify quality parameters for effective provisioning of cloud resources. This method uses Log file for identification of SMI parameters. Log data is used for identifying the requirement of the user. Data analysis is used to identify the parameters. The user can also provide the workload details. Certain quality parameters include bandwidth, reliability, availability, usability, correctness etc..

In[3]K.S Guruprakash and Siva Sathya.s proposed Web Log Analytics framework for cloud service recommendation and prediction. It uses web usage logs. Big data technique is used for analyzing log file because the size may be large. Here some SMI(service measurement index) parameters are taken into account. A mapping can be obtained by extracting SMI parameters from the weblogs.

*b) Future requirement prediction*

There have been some research papers[6][5] based on predicting future requirement values. Samuel.Ajila[13] has proposed a method for predicting cloud resource provisioning. A machine learning technique is used here. The objective of the method is to meet the SLA requirements. VM has to provision in minutes so future resources have to be considered. In this paper, SVM, LR(linear recursion), NN(neural network) are used.

B.v.v.s Prasad[5] has used time bounded, the hour bounded method for predicting future resource requirement. The resource access log file has taken and the requirements are predicted from past log files.

*c) Game Theoretic Resources Management in Cloud.*

Xin Xu and Huiqun Yu have proposed[4] a game theory algorithm for fair resource allocation. They focus on the issues of resource utilization. They worked on the concept of Nash equilibrium to efficiently achieve fair resource allocation among users. The resources are allocated in such a way that the number of resource fragments is minimized. They show that the proposed game theory algorithm has Nash equilibrium and supports fair resource allocation for users. However, most of them just investigate resources of a single cloud service provider. Their work has some drawbacks as they do not consider users requirements and future requirements of the user.

Our work considers resources from multiple service providers and resource are allocated on the basis of predicted requirements of the user.

## III. PROPOSED SYSTEM

### A. Requirement prediction system

The webserver log file is taken for requirement prediction. The log file contains plenty of data such as client IP address, request date/time, page requested, HTTP code, bytes served. These data are identified from log file using Linux commands. The parameters such as reliability, availability, security etc. are taken from the log file by analysing the log file using Linux commands. The parameters are categorized so that these can access easily. All the inconsistencies are removed and some clustering of data is performed. The existing method given in paper[2] is the basis for our work.

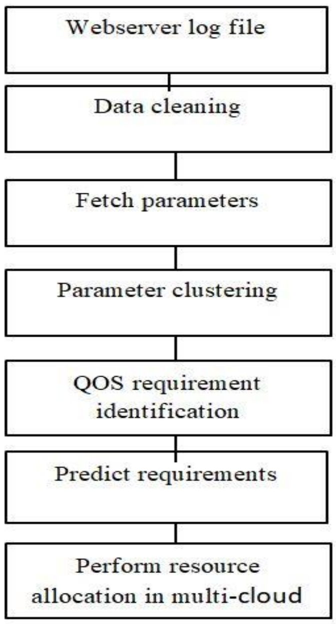


**Figure 1**: requirement prediction

### a. Data cleaning

The log file is taken and data cleaning is performed on the huge data. We used bash commands such as grep to analyze the file and find the data which we want. Necessary data is identified from the log file.

### b. Fetch parameters

The QOS parameters are identified from the data. This will serve as a matrix for evaluating the resource allocation. The QOS parameters include security, reliability, bandwidth, memory etc.

**c. Parameter clustering**

The identified parameters are classified so that it can be accessed easily. We clustered the parameters on the basis of IP addresses.

**d. QOS identification**

From the file, The QOS requirements are identified. A requirement file(CSV format) is created which serves as an input for the future requirement prediction. The cloud provider can use the identified requirement as a matrix for evaluating the resource allocation.

**f. Predict future requirements**

The prediction algorithm is performed on the requirement files and the future requirements are find out. We used SVR and SVM algorithms to predict the future requirements. This phase will be served as an input for the game theory based resource allocation. All requirements can not be predicted. There are some requirements such as storage, which has to be manually taken from user. Also some parameters are manually taken from users. The parameters which are taken for prediction is as follows

1. Reliability: It is the time taken after survival of a system failure
2. Serviceability: It is related to the maintenance such as software update etc.
3. Usability: It is the ratio between the successful operations out of total number of operations
4. Correctness: It is the degree of accuracy provided to users
5. Bandwidth : usually measured using number of bits transferred per unit time for a particular task
6. Computational capacity: It is the measure computed using the ratio between the actual usage time and the expected usage time.
7. Availability: it is the recovery stage of the system in case of system failure.

Apart from the above some parameters are manually taken from the users which are as follows:

1. Storage: Since cloud provides different configurations of servers this parameter can not be predicted. The packages provided to the user varies cloud provider to cloud provider. User has to manually specify the storage type he wants.
2. Cost: The cost is proportional to the storage size. As storage increases the cost also will increases.
3. Memory: This parameter has to be manually taken from user.

**g. Perform resource allocation in multi-cloud**

This is the final phase of the proposed framework. As a result of prediction algorithm, one predicted requirement file is created. This phase will serve as an input for resource allocation using game theory approach

The predicted requirements + input taken from user = input for resource allocation.

**B. Future requirement prediction algorithm**

Input: requirement file
output: future resource prediction values(FRPV)
PM : Physical Machine, REF : user requirement file
foreach pm in cloud do
REF1 ← get requirement file 1
REF2 ← get requirement file 2
PRUM ← do Analysis/Create prediction model

ALG ← Choose the algorithm to be performed on the prediction model

PRFR ← predict Future Requirements (PRFR)
if(PRFR <= Available resources) setOverFlowFlag()
else setUnderFlowFlag()
wish List.addToWishlist(PRUM)
end foreach;
FRPV = do Migration(wishList)
return FRPV
End

The SVR (support vector regression) and SVM (support vector machine) algorithms are performed on the requirement files which are already created from the webserver log files. We have implemented the proposed algorithm and this proposed algorithm can able to perform the following

1. Upload requirement files in csv format
2. Select algorithm to be applied (svr,svm)
3. Create prediction model.
4. View model.
5. Append data.
6. Predict future requirements.

## IV. Resource Management System

There are a number of cloud providers are available and the most popular one is amazon. We have considered multiple cloud service providers. Cloud provider has a large- scaled and distributed datacentre with multiple physical servers and provides physical resources on a payment basis. The virtual machines are created by using VMware and this scenario works by using the software hypervisor. Cloud users deploy their applications on VMs to run their jobs or tasks which are called jobs in our work. Cloud provider offer a group of VMs to the user in which the user can deploy their applications. Each VMs varies in size, the storage size, and the quantities of other resources. User's strategies change over time so cloud providers have to manage the resource allocation decision dynamically. For all these problems, we designed a resource management system. The proposed cloud resource management system is shown in figure 1. This resource management system has three components which include the cloudlet, cloud information service (CIS), cloud broker. The four mentioned components are described as follows.

(i) Cloudlet: this specifies the load of task submitted by the user. Cloudlet contains the configuration of applications or simply the requirements of the user.

(ii) CIS: Defines cloud information service. The Physical server registers their information in CIS. Cloud broker is allowed to access CIS.

(iii) Cloud Broker: Cloud broker manages all the requests of users. The broker checks the specification which is needed to execute the task and also checks for the host which is free.

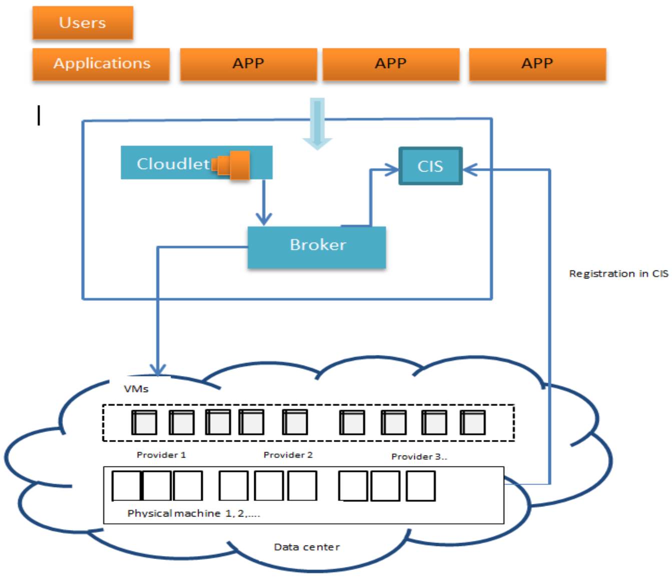


**Figure 2**: A framework of cloud resource management system.

Cloud Broker is the central entity which manages resource allocation decision. The user load is submitted to cloudlet which takes the requirement of the user such as storage size, memory etc. .CIS is the cloud information system stores information such as IP address, memory etc.. of each physical server of cloud service provider. Cloud provider registers its entire physical server in CIS. The Broker has direct communication with CIS. The Broker allocates VM to user according to the requirements in the cloudlet. Broker uses CIS information to identify appropriate VM.

### 4.1 Mathematical Model

Consider there are p physical servers available in a cloud and each is denoted as p, where 1 ≤ p ≤ m. Consider k types of resources available and the resources of each physical server can be defined as follows

$$C^{(p)} = \left(C_1{}^{(p)}, C_2{}^{(p)}, \cdots\cdots C_k{}^{(p)}\right).$$

Where C denotes the capacity vector. The resources are denoted by the letter j. For example (2,4,20) defines that physical server has 2 CPU, 4 GB of Ram and 20 GB of disk storage. The job submitted by user $i$ is denoted as $Ji$, where $i \in \{1, 2, . . . , s\}$.The VMs are predefined by the cloud provider and it is defined by $ri$ =($ri1, ri2, . . . , rij, . . . rik$). The performance of a cloud increases if the number of VMs allocated to each task or user increases.

*Definition 1* (resource requirement matrix). The request made by the user for a resource can be represented as a matrix. Let $R$ be a $s \times k$ dimensional matrix. In the matrix, the rows specify VM types and the columns specify the number of resources.

$$R= \begin{pmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vdots \\ \vec{r}_s \end{pmatrix} = \begin{bmatrix} r_{11} & r_{12} \cdots & r_{1k} \\ r_{21} & r_{22} \cdots & r_{2k} \\ \vdots & \vdots & \vdots \\ r_{s1} & r_{s2} \cdots & r_{sk} \end{bmatrix}$$

The given resource requirement matrix and the capacity vector of physical servers is used to map resources to the cloud users.

*Definition 2* (allocation matrix). An allocation matrix is defined which represents the resource allocation state for the physical server p.

$$A(p)= \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vdots \\ \vec{a}_s \end{pmatrix} = \begin{bmatrix} a_{11^{(p)}} & a_{12^{(p)}} \cdots & a_{1k^{(p)}} \\ a_{21^{(p)}} & a_{22^{(p)}} \cdots & a_{2k^{(p)}} \\ \vdots & \vdots & \vdots \\ a_{s1^{(p)}} & a_{s2^{(p)}} \cdots & a_{sk^{(p)}} \end{bmatrix}$$

Where $a_{ij(p)}$ denotes the number of resource j on physical server p allocated to user i. Each physical server has allocation matrix. The allocation decision will be the set of all allocation states of each physical machine according to user resource requirement matrix.

$$A=\{A^{(2)}, A^{(3)} \dots \dots . A^{(p)}\}$$

The total number of resource allocated to the user denoted as

$$\alpha=\sum aij^{(p)}$$

Where j represent resource and i represent the user.

### 4.2. Game Theoretic Resource Allocation

A cloud provider creates a set of VMs to run the required tasks of user. Each user requests VM to run their job. This would be complex if multiple cloud providers participating towards resource allocation.

In order to allocate VM to the user, we propose an algorithm to manage the resource allocation complexity and to maximize the resource utilization.

### 4.3 Resource Utilization

During the running of the tasks, the resources of the physical server may not be fully used. Suppose physical server 1 can handle 9 tasks at a time and currently it is running 4 tasks. Consider at physical server 2,3 tasks are running. If you observe this physical server 1 fragments are left over. The resources of physical server 1 are not fully utilized.

To address this issue, resource consumption from each physical server is considered.

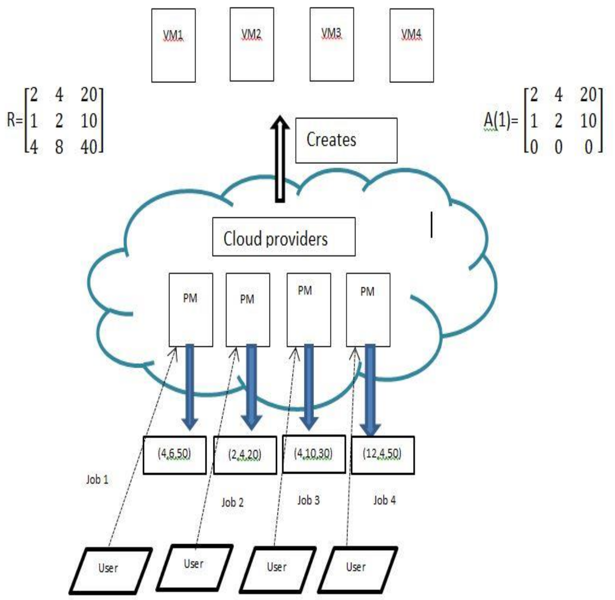


**Figure 3**: illustrates resource allocation

Figure 2 represents resource allocation in the cloud. Cloud providers having a set of heterogeneous physical machines, which altogether can be referred to as a datacentre. These physical machines process a capacity vector. For example (2,12,60) defines a physical machine of 2 CPU, 12 GB memory, 60 GB disk storage. User requests (job) are submitted to the cloud provider. Cloud provider creates Virtual machines that will be allocated to the user. A requirement matrix is created, which shows the available resources in the cloud. An allocation matrix is created, which shows the allocation decision based on the requirement matrix. Every physical server has an allocation matrix. An allocation decision is the combination of the entire allocation matrix.

In our work, the resource allocation problem is modelled as a game with perfect information. Each physical server is treated as a game player and each player poses allocation matrixes.

**(1) Resource Allocation Game**

A resource allocation game is represented as a four-tuple vector $G$ = (P,,,).

(i) $P$ is the players in the allocation game.

(ii) $R$ represents the resource requirement matrix of each user.

(iii) S denotes the strategies of each player.

(iv) $U$ is the utility function of game players.

The physical servers are represented by the parameter p which is the player in this game. Each server is associated with a capacity vector .The user request is translated into a matrix called resource requirement matrix. A utility function is created and it is associated with each of the game players. The objective is to maximize the objective function or the utility function. Further more each player tries to improve the resource utilization.

Maximize $U^{(p)}$

Subject to$\sum_i \sum_p aij^{(p)} \leq C_j$

The utility function represents the objective of resource utilization. We designed a utility function subject to yield better resource utilization. In our work, the sum of current load and loads from other machines represents the utility function. The utility function is shared among every player. The objective is to maximize the utility function. The proposed algorithm will be able to improve resource utilization. The resource allocation game is defined as the communication between each physical servers to make an allocation matrix. The game can be represented as an extension-form tree. Figure 3 shows the extension-form tree. The tasks can be represented in the form of a tree where n is the number of tasks. For all tasks in the graph (G), find the level (L) of every task where n is the number of tasks, i = 1, 2,...,n. Consider the level of root is one as every child is at the next level of parent and level for current task increases one unit. Every level is represented as a game player. In figure 3, each leaf represents the load or the task. The levels are the game players. In figure 3, physical server1, physical server2, physical server 3 are the game players. The alphabets a,b,c etc. represents the strategies. Each player chooses a strategy in such way that it maximizes the utility function. For example (2,12,60) is a strategy which defines 2 CPU, 12 GB memory, 60 GB of disk storage. Each player is aware of the strategy of different players. The utility function is shared between each player or physical servers. We have designed the resource allocation game on VM level.

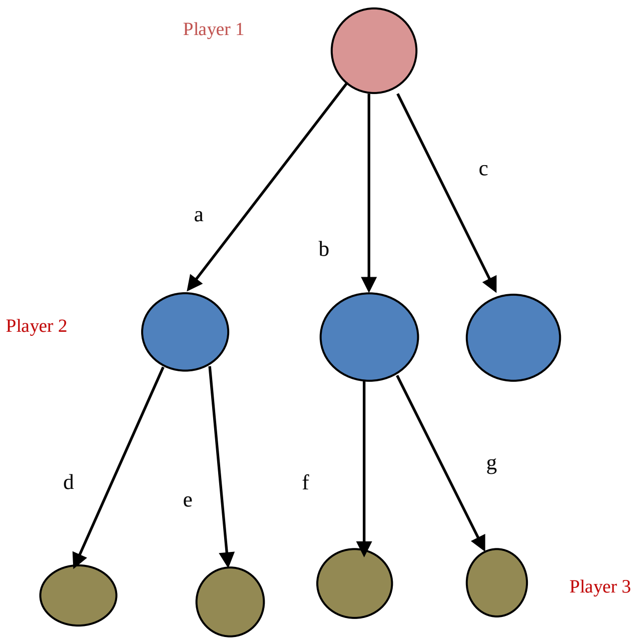


**Figure 4**: The extension-form tree.

### 4.4. Game theory Algorithm

The algorithm for determining the appropriate number of processors based on game theory concept:

//Get the input load/cloudlet input for each tasks

1. For all tasks in the graph (G), find the level (L) of every task where n is the number of tasks, i = 1, 2,...,n. Consider the level of root is one as every child is at the next level of parent and level for current task increases one unit.

//Calculate the machines which have less number of loads (i.e level)

**for each physical server p do**

2. Find the number of tasks in every level and put them in the variable NLi, as L is the number of levels.

**end for**

// Each less loaded machine are labelled as L

3. Consider every level as one player in a way that L = {L1,L2,...,LL}, 1 ≤ i ≤ n.

**for each physical server p do**

4. Consider that S is set of strategies that every player can select. In other words every player can choose one of strategies (number of processors) in a way that the range is from 1 to Nmax S = {1, 2,...,Nmax}.

**end for**

// Calculate the utility or capability for the machines (G), that is sum of current load and the running load

5. Calculating utility for each machine or processor, that for every player (level) is as follow:

U(Gi) =Current load (Li) + load from other machines (Gi)

Check whether utility is less than the strategy, the task is shifted to the machine which have the utility.

6. The best answer will obtain if U(Gi) ≥ S(Gi).

// Update the machine loads after shifting

7. Update the machine with the shifted tasks, AvailLoad(Gi)=Current load (Li) + loads from other machines (Gi).

## V. Performance Evaluation

CLOUD Simulation, is a framework for modeling as well as simulating various services and infrastructures on the cloud. CloudSim can provide a virtualised environment to users. It leverages virtualised environments based on the user requirement such as workload which varies with time. It provides a user both the facility to either model the cloud or simulates various services to the intended user. Another feature of CloudSim is, it provides a custom interface which allows the user to implements policies for the VM's. It provides nearly all types of major services like IaaS, PaaS, and SaaS.

We implemented and simulated the proposed game theory based resource allocation model. For the simulation, we have considered a total of 0-10 cloudlets and 0-5 VMs, where each VM has following configuration: RAM, storage, bandwidth. These configurations are varied from 1 VM to another VM. The cloudlets have the characteristics: input size, output size, no.of cloudlet, pes no. The number of cloudlets and VMs are varied from 1 to 10 to evaluate the effect on time, cost and allocation.

Fig. 5 shows the relationship between number of VM and processing cost. Examine cloudlet properties and VM remaining constant, if the VMs increased from 1 to 10 the cost of processing also increased from $10 to $100. The creation of new virtual machine will cause storage and memory costs to be increased.

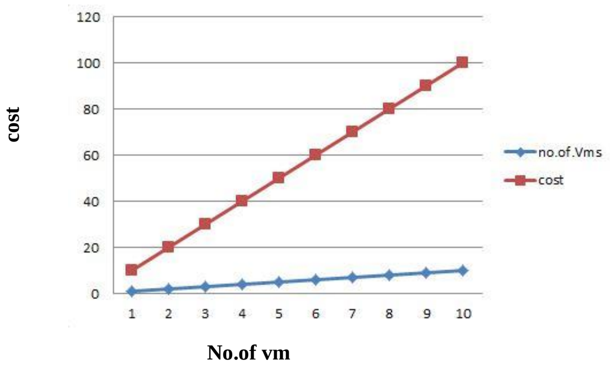


**Fig. 5**. No. of VM's vs Cost.

Fig. 6 shows that Cloudlet length and processing time is proportional to each other. The length of cloudlet increased from 10000 to 100000, the maximum processing time also increased from 100 s to 1000 s. The cloudlet completion time is determined by cloudlet length and VM capacity. If the instruction in a cloudlet is increased, VM will be loaded with more data. This will result more time to execute the cloudlet.

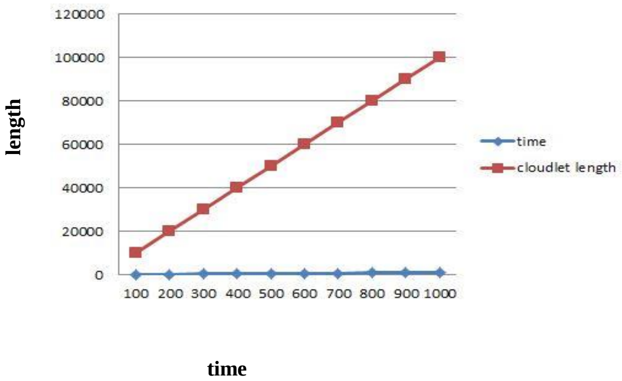


**Fig. 6**. Cloudlet length vs maximum processing time.

Figure 7 shows the results of the best-fit algorithm. We applied the algorithm on the data and the simulated results are given below. Consider there are a total of 10 tasks and 5 VMs are available to run the tasks. Each VMs having different configurations. According to the algorithm, the results are as follows. VM1 runs 4 no.of tasks,VM2 runs 3 no.of tasks and VM3 runs 3 no.of tasks. The remaining VMs are free. The Best suitable hole is allocated to the VM based on the capacity of VM.

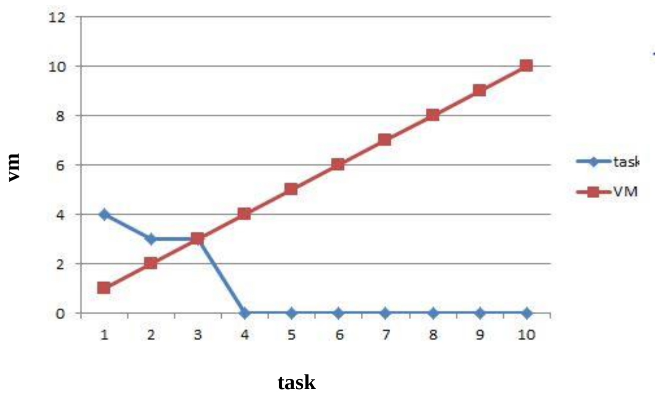


**Fig 7** .Best fit algorithm results

Figure 8 shows the results of proposed game theory algorithm. We simulated the algorithm using a set of sample values. Consider the output of best-fit algorithm (showed in figure 7). The output of game theory algorithm is as follows. The task of Vm2 shifted to VM1 since the capacity of VM1 is 9. So the VM1 runs a total of 7 tasks. The Vm2 will be free and the VM3 runs 3 tasks.

As a result of game theory algorithm, the number of VM involved in the process is reduced when compared to the best-fit algorithm.

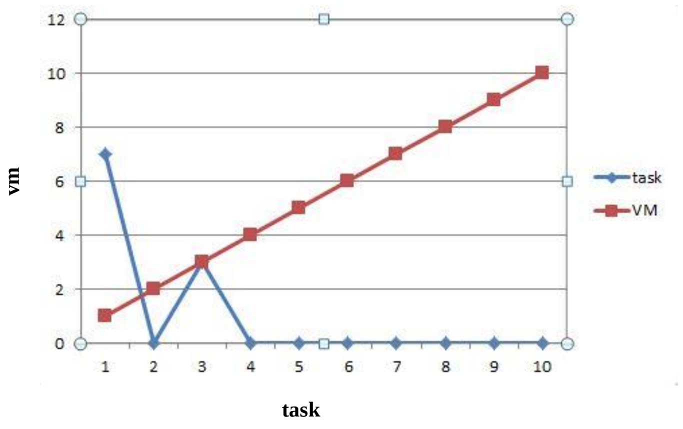


Fig. 8. Game theory algorithm results

## VI. CONCLUSION

In this paper, we carry out research on requirement prediction of the user and the resource allocation problem in cloud computing environment. We consider multi-cloud and as well as multiple types of resources such as storage, memory, and CPU etc. The work operates on a virtual machine to propose an algorithm called game theory. The algorithm supports efficient resource allocation. The implementation results show that the algorithm improves resource utilization. The resource allocation problem is modeled as a finite extensive game.

Simulations and experiments are carried out to analyse the performance of the proposed game theory algorithm by comparing to other algorithms. In the simulation results, it is shown that the proposed algorithm has better performance in resource allocation as well as in resource utilization. The proposed game-theory algorithm can able to perform efficient resource allocation rather than the best-fit algorithm.

## VII. REFERENCES


1 Yu sun a,jules white b, " ROAR: A QoS-oriented modeling framework for automated cloud resource allocation and optimization",june 2016,vol 116

2 K. S. Guruprakash1 and S. Siva Sathya2," Indian Journal of Science and Technology, Vol 9(30), DOI: 10.17485/ijst/2016/v9i30/99011, August 2016.

3 K. S. Guruprakash1 and S. Siva Sathya2," International Journal of Applied Engineering Research ISSN 0973-4562,Number 16 (2015) pp 37770-37776 © Research India Publications. http://www.ripublication.com

4 Xin Xu and Huiqun Yu," A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing", Volume 2014, Article ID 915878,

5 B.V.V.S Prasad,"Predicting Future Resource Requirement for Efficient Resource Management in Cloud", Volume 101– No.15, September 2014

6 Samuel A. Ajila,"Predicting Cloud Resource Provisioning using Machine Learning Techniques", 978-1-4799-0033-6/13/$31.00 ©2013 IEEE