

# Optimized Virtual Resource Deployment using CloudSim

Anesul Mandal

*Software Professional,  
Aptsource Software Pvt. Ltd., A-5 Rishi Tech Park, New Town,  
Rajarhat, Kolkata, India.*

Kamalesh Karmakar

*Assistant Professor,  
Department of Computer Science & Engineering and Information Technology,  
Meghnad Saha Institute of Technology, Kolkata, India.*

**Abstract-** Cloud Computing provides reliable, secure, fault-tolerant, sustainable, and scalable infrastructures services to customer. It is becoming the default mode of services to the small and large organization. Deployment of resources to the cloud environment is a challenging research area. A number of research efforts are published to optimize the deployment of cloud resources. CloudSim is one of the tools that help optimize the resource allocation in cloud environment. Present work enhances the features of CloudSim to optimize the cost of resource deployment based on reusing the idle time of the rented VMs.

**Keywords –** CloudSim, Cost Optimization, Satisfaction Level, VM Deployment, VM Performance

## I. INTRODUCTION

Cloud Resource Management is very crucial task for cost minimization. In this paper different strategies are being described to optimize VM deployment for processing jobs in Cloud Environment. As in real time environment VM deployment is associated with huge Infrastructural cost, here CloudSim is being used to simulate network, its configuration and deployment of VMs in Cloud Environment. CloudSim is a framework developed by the GRIDS laboratory of University of Melbourne which is used for seamless modeling, simulation and experimenting on designing Cloud computing infrastructures. It is a self-contained platform for modeling data centers, service brokers, scheduling, and allocation policies. It has the flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services.

In every proposed strategy VM deployment cost is calculated for processing a set of jobs. But till now idle time of a VM is not being reused. In these proposed strategies idle time of a VM is reused to process another job without allocating a new VM for the newly arrived job. As VMs are deployed by users directly in a Cloud Service Provider's (CSP) infrastructure, these idle times cannot be managed and traced and more over the CSP cannot reuse it for other purpose as it is violation of Service Level Agreement. Here we are going to introduce a broker who can act as an intermediary to provide the service in lower cost by optimizing the resource usage. So users will access Cloud Infrastructure via Broker API. Broker reserves VMs from service provider according to users' need. Customers reserve VMs from the Broker. Since Broker can reuse the free time of the VM, they can provide the same service with lower cost compared to the actual service provider. In this concept both the Broker and the Customer are benefited.

In this context section 2 describes the related work till now and section 3 describes the proposed algorithm in details followed by experimental results in section 4.

## II. RELATED WORK

Cost optimized resource allocation in cloud environment is a challenging factor. Quite a good amount of research work is carried out on resource allocation in cloud environment. Some software tools are used to decide the optimal resource deployment and evaluate the performance in cloud. Following are some of the concepts that are used for optimal resource deployment...

- Track the performance of all the VMs like CPU Usage, Memory Usage, Bandwidth Usage and so on and decide to scale up and down. This concept reduces the cost. [1]

- A cloud service broker is designed to manage and monitor VMs in cloud environment and decides when resource should be migrated or scaled to reduce cost. [2]
- A parallel processing framework based on Nephele manages the resources automatically while executing jobs and allocates the VMs when needed and de-allocates when it completes its work or is not used for a long time. [3]
- A Hybrid Fuzzy clustering algorithm assigns appropriate resource to task that exactly satisfy its needs. This algorithm avoids powerful resources to simple and medium scale tasks and poor resources to complex and large scale tasks. This is efficient resource management in cloud. [4]
- A load balancing algorithm manages a resource allocation table from where a least used and least loaded VM is returned to Datacenter Controller. For this a popular VM policy is used. [5]

This research work simulates a cloud environment and decides optimal set of VMs which gives better performance by using different strategies. In this context next section discusses about the proposed concept.

### III. PROPOSED ALGORITHM

Resource optimization in cloud environment is required to minimize the cost. Now-a-days a number of open source tools are available in the market that helps achieve this solution. This work takes a further step ahead to more reduce the cost. The Cloud Service Providers charge for the allocated VMs based on its configuration and reserved time period. That means for processing a set of jobs the VM configuration “ON\_DEMAND\_LARGE” will provide a cost which is different from “ON\_DEMAND\_XLARGE” or “ON\_DEMAND\_DEFAULT”. If a VM finishes its work, it remains idle. This leads to underutilization of the reserved VMs. This research work provides an idea how to reuse the free time of the reserved VMs and compares the cost for different strategies.

#### A. Problem Definition–

Optimize cost by optimizing the deployment of VMs by simulating a cloud environment using CloudSim. By changing the configuration of the VMs and reusing the free time of the VMs cost is calculated. This process is carried out repeatedly until an optimal configuration of the VMs are achieved that minimizes the cost.

#### B. Reuse of Idle Time of VM –

Cost is calculated based on the configuration of the VM. This research considers the configuration like hypervisor used, RAM, number of processors, MIPS, bandwidth and so on. It decides the optimal number of VMs by changing the configuration and reusing the idle time of the VMs for a set of jobs. A pool of available idle VMs is maintained. Every time a job is ready to be allocated to a VM, the pool of VMs is searched to find out a suitable VM for processing the job. If a job can be allocated to the free time of a VM then the cost of processing that job to that VM will be 0. The cost and number of VMs required are noted. This work uses two algorithms for selecting the VMs from free pool of VMs. One is Firstfit and another is Bestfit.

#### C. Firstfit Algorithm –

The Firstfit algorithm selects a VM from the free pool of VMs, the first available VM which has enough time to process the job. The VM selection algorithm from free pool of VMs is described below.

Process 1:

Loop <All Jobs>

Pickup (Job)<sub>i</sub> from set of submitted jobs.

(Job)<sub>i</sub>\_Status = New

Calculate expected Processing Time of (Job)<sub>i</sub>.  $(EPT)_i = (\text{Job Length})_i / \text{MIPS of (VM)}_i$

Loop <Free VM Pool>

Calculate available free time of (VM)<sub>j</sub>.  $(FreeTime)_j = (\text{Reserved End Time})_j - \text{Current Time}$

If  $(FreeTime)_j \geq (EPT)_i$  )

<If Block>

Allocate (Job)<sub>i</sub> to (VM)<sub>j</sub>

```

                (Job)i_Status = Processing
                Remove (VM)j from Free VM Pool
            <End If Block>
        End Loop <Free VM Pool>
        If( (Job)i_Status == New )
            <If Block>
                Reserve a new VM (VM)k
                Allocate (Job)i to (VM)k
                (Job)i_Status = Processing
            <End If Block>
        End Loop <All Jobs>

```

Process 2:

```

        If ( (Job)i_Status == Finished )
            <If Block>
                If (Reserved End Time of (VM)j > Current Time)
                    Send the (VM)j to Free Pool of VMs
                Else
                    Remove (VM)j from deployment
            <End If Block>

```

#### D. Bestfit Algorithm –

The Bestfit algorithm selects the most suitable VM whose free time is nearest to the expected execution time of the job. The VM selection from free pool based on this algorithm is given below.

Process 1:

```

        Loop <All Jobs>
            Pickup (Job)i from set of submitted jobs.
            (Job)i_Status = New
            Selected_VM = -1
            Selected_FreeTime = <Some big number>
            Calculate expected Processing Time of (Job)i. (EPT)i = (Job Length)i / MIPS of (VM)i
            Loop <Free VM Pool>
                Calculate available free time of (VM)j. (FreeTime)j = (Reserved End Time)j – Current Time
                If ( (FreeTime)j >= (EPT)i )
                    <If Block>
                        If ( (FreeTime)j < Selected_FreeTime )
                            <If Block>
                                Selected_VM = (VM)j
                                Selected_FreeTime = (FreeTime)j

```

```

                                <End If Block>
                        <End If Block>
    End Loop <Free VM Pool>
    If (Selected_VM != -1)
        <If Block>
            Allocate (Job)i to Selected_VM
            (Job)i_Status = Processing
            Remove Selected_VM from Free VM Pool
        <End If Block>
    Else
        <Else Block>
            Reserve a new VM. (VM)k
            Allocate (Job)i to (VM)k
            (Job)i_Status = Processing
        <End Else Block>
    <End If Block>
End Loop <All Jobs>

```

Process 2:

```

If ( (Job)i_Status == Finished)
    <If Block>
        If ( Reserved End Time of (VM)j > Current Time)
            Send the (VM)j to Free Pool of VMs
        Else
            Remove (VM)j from deployment
        <End If Block>

```

Another process will check the free pool from time to time and remove the expired VM. Next section shows the experimental results by processing a set of jobs in a simulated environment.

#### IV. EXPERIMENT AND RESULT

Jobs that are considered for processing are in Standard Workload Format (SWF). Jobs are taken from two swf files CTC-SP2-1996-3.1-cln.swf and RICC-2010-1.swf which contains jobs of different size. Each job is submitted for processing at different time.

##### A. Job Processing With Existing CloudSim –

A number of set of jobs are submitted. Each job is assigned to a new VM. No VM is reused though it has enough free time to allocate next job. The cost and number of VM required are noted. The results are plotted in a bar chart. VM can be reserved for multiple of a unit time (usually 1 hour). CloudSim decides number of VMs and time needed to reserve for a set of jobs. Based on this concept total cost is calculated.

**B. Job Processing With Reuse of VM –**

A number of set of jobs are submitted. A pool of available VMs is maintained. Every time a job is ready to be allocated to a VM, the pool of VMs is searched to find out a suitable VM for processing the job. If a job can be allocated to the free time of a VM then the cost of processing that job with that VM will be 0. The cost and number of VMs required are noted. The results are plotted in a bar chart. There are two algorithms for selection of a free VM from the pool of free available VMs. These are Firstfit and Bestfit algorithms.

*First Fit Algorithm*

This algorithm selects a VM from the pool of available VMs. The first free VM which has enough free time to process the job, is selected.

*Best Fit Algorithm*

This algorithm selects a VM from the pool of available VMs. The most suitable VM whose free time is nearest to the expected execution time of the job, is selected.

*Number of VMs comparison*

The table in Figure 1 and the graph in Figure 2 show the comparison of number of VMs required for different set of jobs for existing CloudSim and reuse of VMs for both Firstfit and Bestfit algorithms.

No. of Jobs	100	200	300	400	500	600	700	800
No Reuse	100	200	300	400	500	600	700	800
Reuse Firstfit	39	63	85	108	132	146	177	208
Reuse Bestfit	39	63	85	108	132	146	177	208

900	1000	5000	10000	15000	20000	25000	30000	35000
900	1000	5000	10000	15000	20000	25000	30000	35000
230	253	1130	2062	3201	4420	5571	6471	7424
230	253	1130	2062	3200	4419	5570	6470	7423

Figure 1. Comparison of No. of VMs

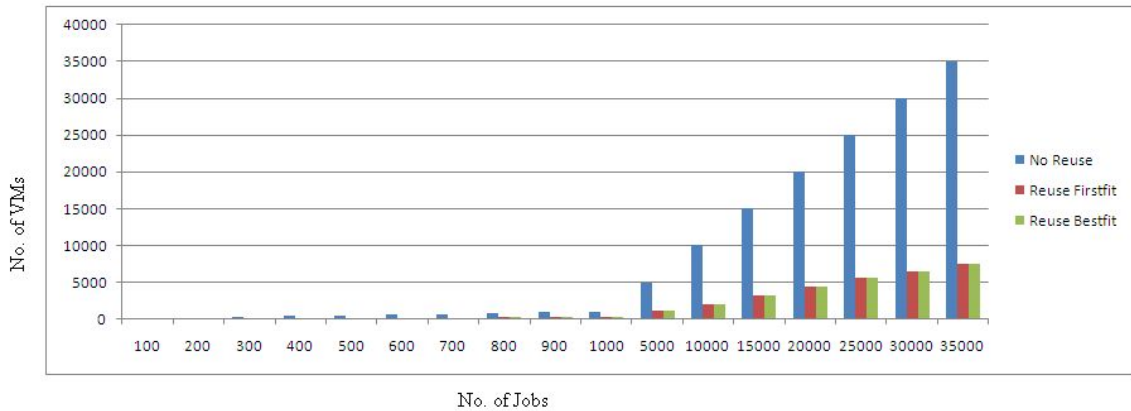


Figure 2. Comparison of No. of VMs

The graph shows Total No. of VMs required against Number of Jobs for No Reuse of VM, Reuse Firstfit and Reuse Bestfit. As the number of jobs increases total No. of VMs also increases linearly. The No. of VMs is always greater for no reuse of VM than that of Firstfit or Bestfit. But that for Firstfit and Bestfit is almost same. There is a little variation.

*Cost Comparison*

The table in Figure 3 and the graph in Figure 4 show the comparison of Cost for different set of jobs for existing CloudSim and reuse of VMs for both Firstfit and Bestfit algorithms.

No. of Jobs	100	200	300	400	500	600	700	800
No Reuse	10.03	18.87	27.37	36.81	45.31	53.81	62.82	71.57
Reuse Firstfit	4.85	7.23	9.1	11.99	14.03	15.22	18.36	21.25
Reuse Bestfit	4.85	7.23	9.1	11.99	14.03	15.22	18.36	21.25

900	1000	5000	10000	15000	20000	25000	30000	35000
80.16	90.53	446.68	893.95	1347.59	1796.05	2252.67	2696.88	3141.86
23.21	27.03	117.73	219.22	344.68	471.75	601.21	696.92	797.9
23.21	27.03	117.73	219.22	344.59	471.67	601.12	696.83	797.81

Figure 3. Comparison of Cost

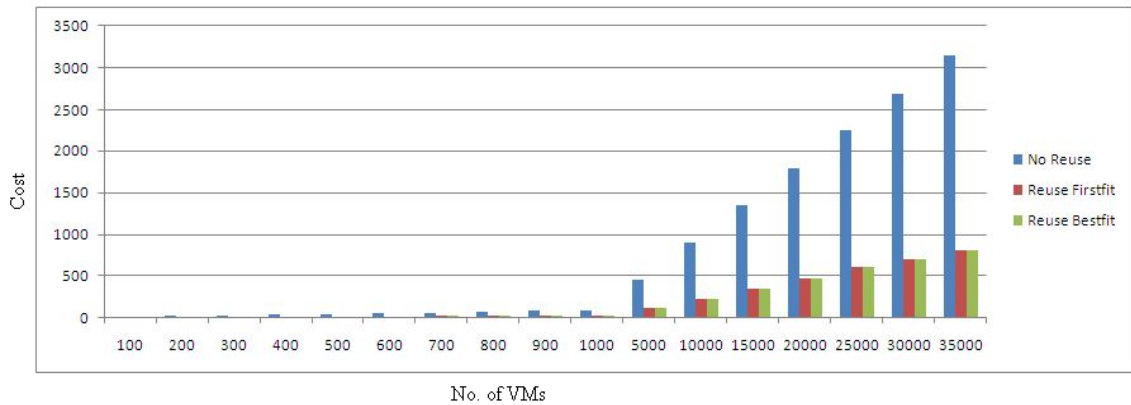


Figure 4. Comparison of Cost

The graph shows Total Cost against Number of Jobs for No Reuse of VM, Reuse Firstfit and Reuse Bestfit. As the number of jobs increases Cost also increases linearly. The Cost is always greater for no reuse of VM than that of Firstfit or Bestfit. But the cost for Firstfit and Bestfit is almost same. There is a little variation.

*No. of Jobs Vs. VM Difference*

The table in Figure 5 and the graph in Figure 6 show comparison between No Reuse and Reuse Firstfit and Reuse Bestfit. This also shows comparison between Firstfit and Bestfit.

No. of Jobs	100	200	300	400	500	600	700	800
No Reuse - Firstfit	61	137	215	292	368	454	523	592
No Reuse - Bestfit	61	137	215	292	368	454	523	592
Firstfit - Bestfit	0	0	0	0	0	0	0	0

900	1000	5000	10000	15000	20000	25000	30000	35000
670	747	3870	7938	11799	15580	19429	23529	27576
670	747	3870	7938	11800	15581	19430	23530	27577
0	0	0	0	1	1	1	1	1

Figure 5. Difference of No. of VMs against No. of Jobs

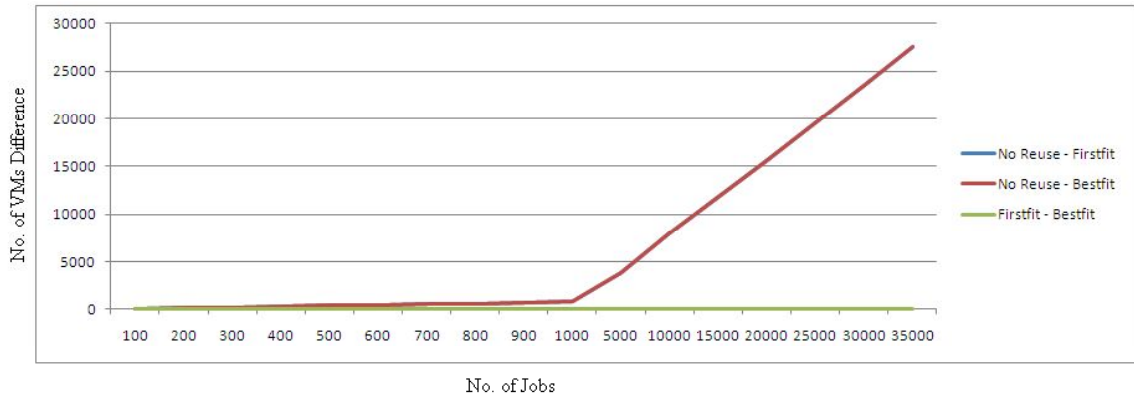


Figure 6. Difference of No. of VMs against No. of Jobs

The graph in Figure 6 depicts that number of VMs for No Reuse and Reuse does not have any notable difference when the numbers of jobs are less but for higher number of jobs this difference increases steeply.

The table in Figure 7 and graph in Figure 8 compare between Firstfit and Bestfit more closely. It is clear from the graph that for less number of jobs Firstfit and Bestfit does not shows any remarkable difference. But as the number of jobs increases Bestfit always shows better performance than Firstfit.

No. of Jobs	100	200	300	400	500	600	700	800
Firstfit - Bestfit	0	0	0	0	0	0	0	0

900	1000	5000	10000	15000	20000	25000	30000	35000
0	0	0	0	1	1	1	1	1

Figure 7. Comparison of Firstfit and Bestfit

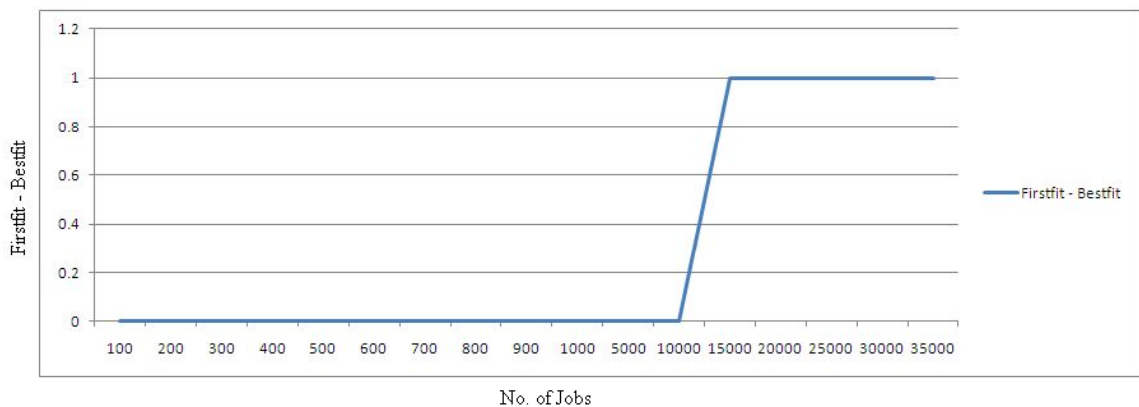


Figure 8. Difference of No. of VMs against No. of Jobs



### *Cost Optimization*

At the provider side, simulation environment allows evaluation of different kinds of resource leasing scenarios under varying load and pricing distributions. Such studies could aid the providers in optimizing the resource access cost with focus on improving profits. In the absence of such simulation platforms, cloud customers and providers have to rely either on theoretical and imprecise evaluations, or on try-and-error approaches that lead to inefficient service performance and revenue generation.

This research work optimizes based on configuration of the VM and its reuse. This means that change the configuration of VM and process the set of jobs. These steps are repeated for number of times and finally an optimal configuration is decided.

The relationship of Service Provider with Customer is one-to-many. Between customer and service provider there is another party called Broker. In real scenario the customer reserves a time slot from the service provider. The provider charges the customer for the whole reserved period. Customer uses the services. His job may be finished before the reserved time of a VM expires. This free time of a VM is wastage. Customer does not use this free time. So the Broker comes into picture. The Broker reserves a time slot for some resources from Service Provider. Customer communicates with the broker and reserves a time slot for a VM. When the customer job is finished, the broker uses the free time of the resources. The broker can re-allocate the free time to another job. So, the broker is paying to the Service Provider once, but he can use the slot for multiple customers. This reduces the cost. So, broker can offer services to the customer with cheap rate.

Our goal is to optimize the use of cloud resources so that the cost can be minimized. Both customer and the broker will be benefited in this concept. Different algorithms can be used for optimal use of the free time. We used different algorithm for job allocation to VM's free time and analyzed the performance.

Once a VM processed a job, the VM becomes free. The main goal of this research work is to reuse this free available time of a VM by applying different VM selection algorithm. The present work tries to assign the next coming jobs to this VM. If that VM can be used then its processing cost for that job will be 0. So, the VM is reserved only once but can be used for multiple customers and multiple jobs. This concept optimizes the cost based on reuse and configuration.

### *Satisfaction Level and Delay Factor*

User wants to finish the job as soon as possible and also wants to process them with minimum cost. So user satisfaction can be defined in two ways:

- (1) How User is satisfied based on completion time of a set of jobs.
- (2) How User is satisfied based on cost.

This paper considers user satisfaction based on time of completion and optimizes the cost. Satisfaction Level is defined with the help of delay time for processing the jobs. If user wants high satisfaction then delay time is low and vice versa. This means that if user wants more satisfaction then less delay time will be introduced to start processing the job and if user satisfaction is less then more delay will be introduced to start processing the job. This delay time increases the chance of reusing the free time of the VM. If no delay is introduced then job will be allocated to a free VM if available otherwise a new VM will be allocated as soon as it arrives. On the contrary if a delay is introduced then if no free VM is available then the job is sent to a waiting queue. In every tick (a second or a millisecond as an example) the delay time of all the jobs in waiting queue is reduced by one time unit and a job whose remaining delay time is minimum, is selected from the queue and tries to allocate in a free VM. This process is repeated until the delay time becomes 0 and that job is allocated to a new VM. So high satisfaction level means less reuse of VM and low satisfaction means more reuse of VM.

$$df = 100\% - SL \quad \text{where } df \rightarrow \text{Delay Factor, } SL \rightarrow \text{Satisfaction Level}$$

$$dt = eXt * df \quad \text{where } dt \rightarrow \text{Delay Time, } eXt \rightarrow \text{Expected Execution Time}$$

### *Satisfaction Level Vs. No. of VM Used and its Cost*

10000 jobs have been processed for different satisfaction level. The table in Figure 9 shows the result of processing 10000 jobs for different satisfaction level using the Firstfit and Bestfit algorithms. The result shows that number of VMs required and Cost are directly proportional to Satisfaction Level and inversely proportional to the Delay Factor.

Satisfaction Level (SL)(%)	Delay Factor (df)	No. of VMs		Cost	
		Firstfit	Bestfit	Firstfit	Bestfit
0	1.0	1768	1764	194.225	193.885
10	0.9	1800	1801	196.945	197.030
20	0.8	1838	1839	200.175	200.260
30	0.7	1878	1878	203.575	203.575
40	0.6	1888	1888	204.425	204.425
50	0.5	1914	1913	206.635	206.550
60	0.4	1933	1933	208.250	208.250
70	0.3	1949	1951	209.610	209.780
80	0.2	1972	1973	211.565	211.650
90	0.1	2018	2018	215.475	215.475
100	0.0	2062	2062	219.215	219.215

Figure 9. Satisfaction Levels

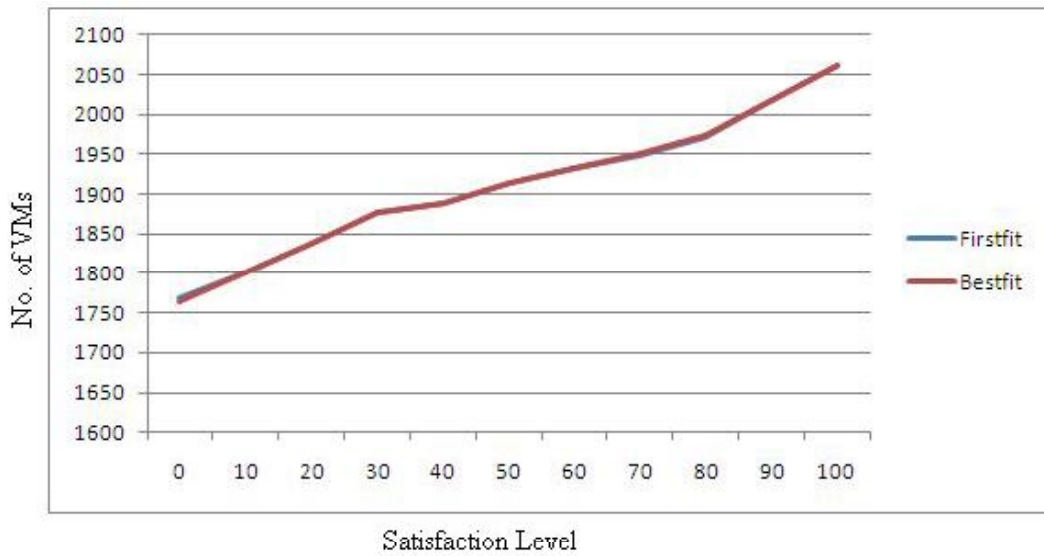


Figure 10. Satisfaction Level Vs. No. of VMs

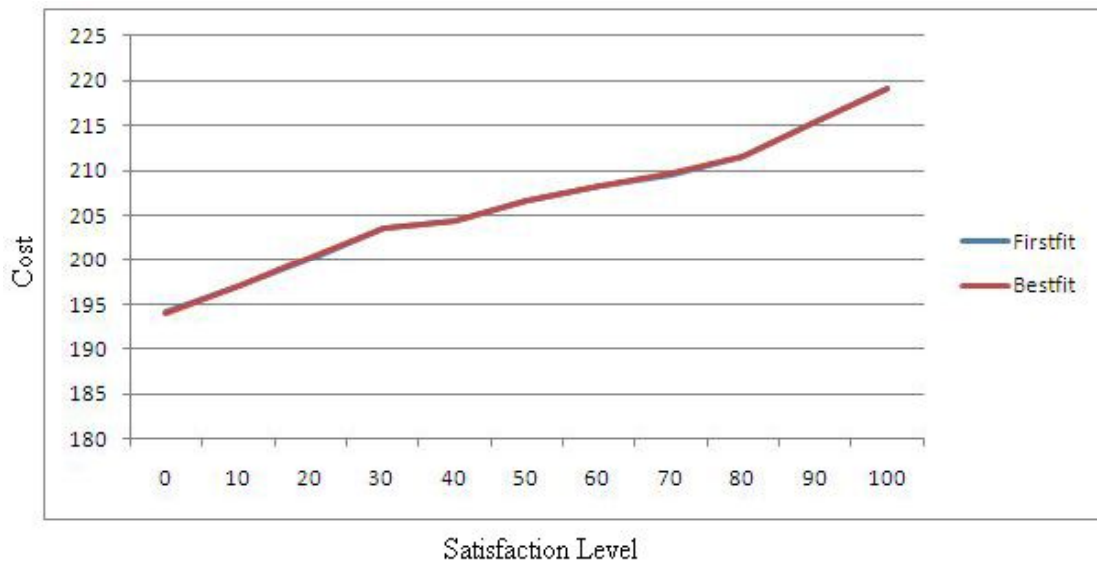


Figure 11. Satisfaction Level Vs. Cost

### V.CONCLUSION

VM deployment strategies have been proposed utilizing free time of Virtual Machine Resources. Here user requests are processed considering different satisfaction level. When user reserved resources become free, without de-allocating the VM from service provider, it is being maintained in the free resource pool for remaining reserved time. These VMs are being used for processing new jobs, if the job can be executed in this free remaining time. This developed component works as a broker for Cloud Service Providers.

The present work enhances the existing features of an open source tool to reduce the cost. It is obvious that the present work provides better performance compared to existing one. There are a lot of scope exist for future work. For instance new algorithm for VM selection from free pool can be suggested. Another may be creation of relationship of

VM configuration and Job length. That means which configuration is suitable for which job size. Present work considers all the VMs are of same configuration. So, the third scope may be use of VMs of different configuration so that simple job will be allocated to simple VM and complex job to powerful VM.

#### REFERENCES

- [1] Kamalesh Karmakar, Anesul Mandal. "Cost Optimized Virtual Machine Deployment in Eucalyptus for Autonomous Systems". International Journal of Innovations in Engineering and Technology, Volume 3, Issue 4, April-2014.
- [2] Kamalesh Karmakar, Tamal Mukherjee. "Virtual Machine Monitoring & Management in Cloud Environment". International Journal of Scientific & Engineering Research, Volume 5, Issue 1, January-2014.
- [3] Dr.S.Sujatha, R.Chinnasamy. "Efficient Dynamic Resource Allocation Using Nephele in a Cloud Environment V.Praveenkumar". International Journal of Scientific & Engineering Research, Volume 3, Issue 8, August-2012.
- [4] k.Sindhuja, P.Anand Raj. "Efficient Framework For Resource Management In Cloud Using Hybrid Fuzzy Clustering". International Journal of Scientific & Engineering Research, Volume 5, Issue 3, March-2014.
- [5] Vartta Siyal, Dr. Naveen Choudhary, Dr. Dharm Singh. "Efficient Load Distribution of VM in Cloud Computing". International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013.