I J C T A, 9(41) 2016, pp. 1103-1111 ISSN: 0974-5572
© International Science Press

ACTIVITY BASED COST SCHEDULING USING PRIORITY APPROACH IN CLOUD COMPUTING

Monika, Pardeep Kumar and Sanjay Tyagi

Abstract: Since cost is the most effective factor in cloud computing, so in this paper, an approach is being proposed based on cost applied on datacenters using basic approach of ABC (Activity-based Cost) scheduling algorithm. The proposed approach works in two phases. In the first phase, we categorized three datacenters' based on cost. Tasks were assigned to datacenters based on cost. In the second phase, three different priority queues were implemented and tasks were assigned to VMs' according to their priorities.

Key Words: *Activity Based Cost approach, Cloud Computing, Datacenter, Datacenter Based Approach, Scheduling Algorithm*

I. INTRODUCTION

Before cloud computing, resources were used in a limited way and the cost of hardware and their maintenance were also high. When cloud computing came in existence, cost of maintaining resources became less. That's why researchers are taking more attention in this field. Cloud computing's on-demand network access, resource pooling, elasticity, and computing resources availability, e.g., networks, servers, and storage had made it so popular. Every organization is thinking of using these services. As cost is the most impacting factor for an organization, researchers started paying more attention in cost effective scheduling algorithms. ABC is one of the cost effective approaches used in cloud computing.

II. RELATED WORK

Cost is the most considerable part of an organization not only in recent years, but also since when organization came into existence. To reduce, cost Gary Cokins introduced Activity Based Cost approach in management and named "ABC/M" in 1988 [1]. Further, in 1993, it was also implemented by Yair M. Babad et. al. by grouping the activities into single drive to calculate or trace the cost of product or service [2]. This was the starting age of ABC algorithm. Carsten Homburg (2001) proposed algorithm based on ABC [3]. He suggested that with the existing cost drivers (cost of activities that are interconnected or that affect the other activities), a new combination of cost driver should be replaced. With this approach, system complexity with ABC will be same but cost allocation will be accurate.

When cloud computing technology came in existence, researchers also tried to implement ABC approach in scheduling. For this, QI CAO et. al. (2009) implement optimized ABC algorithm [4]. The proposed algorithm used priority in ABC that is calculated on the basis of task cost, individual resource cost and earned profit from individual tasks. Requested tasks were consigned to VMs according to defined priorities. Instead of using three queues, Shachee Parikh et. al. (2010) used double level priority queues in ABC approach of scheduling [5]. In their proposed methodology, all tasks were sorted according to their processing power and provide the resources according to tasks' turnaround time. Priority in ABC was used by Mrs. S. Selvarani et. al. (2010) in a different way [6]. Their proposed work was based on new

Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, India
Emails: monika.banasthali@gmail.com, mittalkuk@gmail.com, tyagikuk@gmail.com

arriving tasks. New tasks were arranged into priority queues according to their size and priority. Ashutosh Ingole et. al. (2011) suggested another cost effective algorithm based on ABC [7]. In their proposed algorithm, firstly highest priority tasks were allotted to the resources and then medium & low priority tasks were shifted and consigned to the resources according to their priorities. Except, ABC approach in cloud computing Zhi Yang et. al. (2011) proposed another cost effective scheduling algorithm [8]. In their proposed algorithm maximum profit was achieved through ordering supplier's need. First supplier's tasks with the highest profit were assigned resources before second highest profit tasks. The process was continued until all needs got the resources.

III. ABC (ACTIVITY BASED COST) APPROACH

Activity based cost approach is used not only for finding the cost of resource, but also performance cost [4] [6] [7]. Activity based approach means each activity is different and cost occurring from these activities should be different. In this approach, cost is measured in terms of CPU, memory and time required while fulfilling the individual request. After calculating the cost, priority is calculated of the individual task and divide these tasks into three categories HIGH, MID, and LOW. The following equation calculates individual task's priority [4] [6] [7]:

$$L_k = \sum_{i=0}^{n} R_{i,k} \; * \; C_{i,k} \; / \; P_k \qquad (1)$$

Here n is the no. of resources. Other parameters are explained below:

1) $R_{i,k}$: The i^{th} individual use of resources by the k^{th} task.
2) $C_{i,k}$: The cost of the i^{th} individual use of resources by the k^{th} task.
3) P_k: The profit earned from the k^{th} task.
4) L_k: The priority level of the k^{th} task.

IV. PROPOSED WORK

Earlier researches were performed at the single datacenter, which causes slow execution speed. All tasks priorities were calculated and divided into different categories according to their priorities. After categorizing, tasks are grouped in three different queues (LOW, MID, HIGH) and are assigned resources simultaneously for the fast execution. In this paper, a new approach is proposed in two phases, based on priority. In the first phase, datacenters were distributed based on their cost. Service provider divides three datacenters and assigns the tasks as per the cost requirement. In the second phase, basic ABC algorithm has been applied on individual host based on task's priority using equation (1). Flowchart of proposed algorithm is as under:

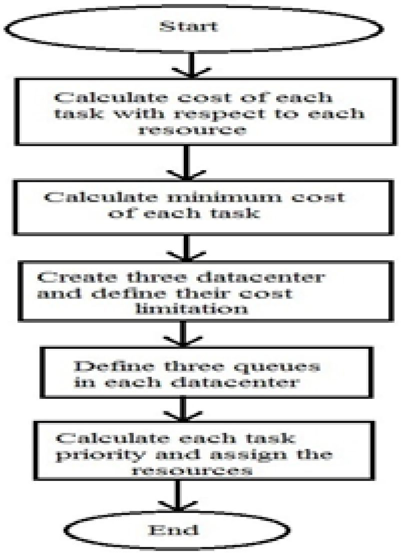

Start
Calculate cost of each task with respect to each resource
Calculate minimum cost of each task
Create three datacenter and define their cost limitation
Define three queues in each datacenter
Calculate each task priority and assign the resources
End

Figure 1 Datacenter Based ABC algorithm

Proposed Algorithm:

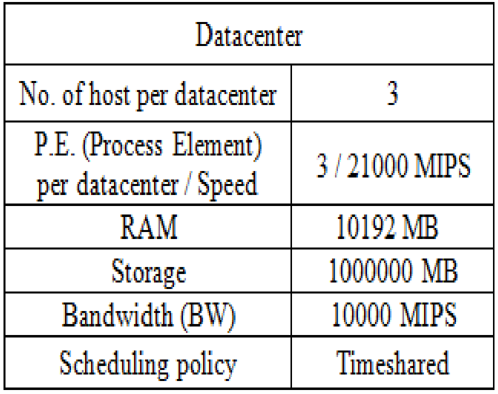
r = resource size, t = task size

1) Calculate ETM (Execution Time Matrix) of each task with respect to each resource by using
   $\sum_{i=0}^{t} \sum_{j=0}^{r}$ t.length / r.mips .
2) Calculate CM (Cost Matrix) of each task with respect to resource by using
   $\sum_{i=0}^{t} \sum_{j=0}^{r}$ etm[i][j] * r.mips .
3) Find out minimum cost of each task w.r.t. resource.
4) Catagorise datacenters according to cost.
5) Calculate priority of each task in each datacenter by using
   task.priority = cost/profit
6) Assign VMs according to their cost and priority.

In proposed approach, the execution time of tasks is calculated based on their MIPS.

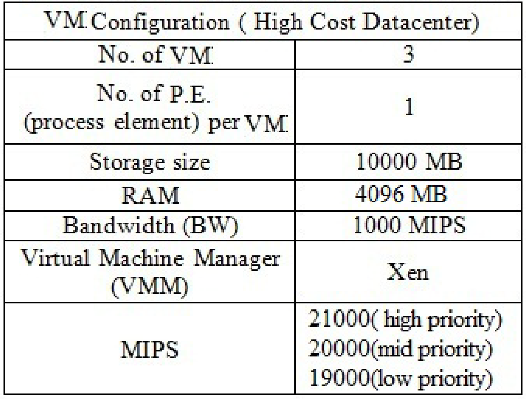
4.1. *Experimental parameters:* For this proposed work, following configuration has been taken in cloudsim simulator:

Table1. Datacenter configuration



Datacenter	
No. of host per datacenter	3
P.E. (Process Element) per datacenter / Speed	3 / 21000 MIPS
RAM	10192 MB
Storage	1000000 MB
Bandwidth (BW)	10000 MIPS
Scheduling policy	Timeshared

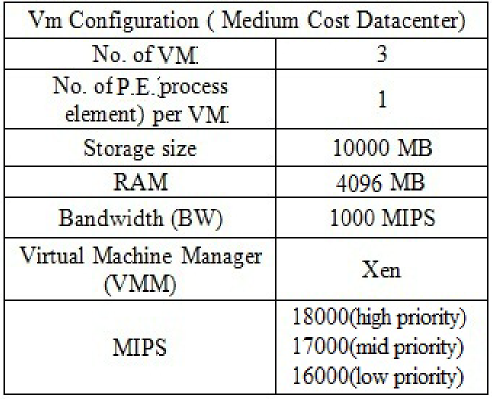
Table 2 VM Configuration for high cost datacenter



VM Configuration (High Cost Datacenter)	
No. of VM	3
No. of P.E. (process element) per VM	1
Storage size	10000 MB
RAM	4096 MB
Bandwidth (BW)	1000 MIPS
Virtual Machine Manager (VMM)	Xen

MIPS	21000(high priority)
	20000(mid priority)
	19000(low priority)

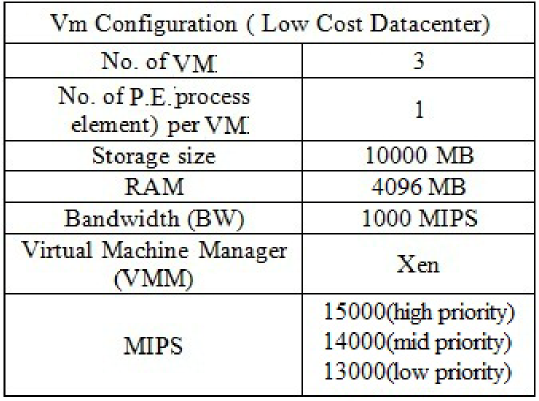
Table 3 VM Configuration for medium cost datacenter



Vm Configuration (Medium Cost Datacenter)	
No. of VM	3
No. of P.E.(process element) per VM	1
Storage size	10000 MB
RAM	4096 MB
Bandwidth (BW)	1000 MIPS
Virtual Machine Manager (VMM)	Xen

MIPS	18000(high priority)
	17000(mid priority)
	16000(low priority)

Table 4 VM Configuration for low cost datacenter

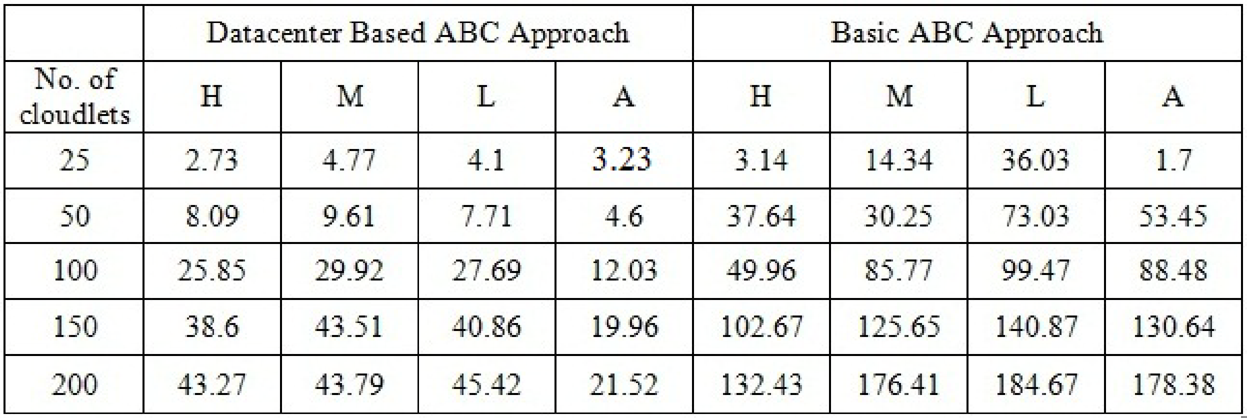


Vm Configuration (Low Cost Datacenter)	
No. of VM	3
No. of P.E.(process element) per VM	1
Storage size	10000 MB
RAM	4096 MB
Bandwidth (BW)	1000 MIPS
Virtual Machine Manager (VMM)	Xen

MIPS	15000(high priority)
	14000(mid priority)
	13000(low priority)

4.2. *Result and Analysis:* After implementing proposed algorithm, tasks were divided among three datacenters' based on cost and then tasks were provided resource or VMs based their priorities. Three VMs with different MIPS were implemented on each datacenter. Our result shows less execution time as compared to basic ABC scheduling policy. Table 5 shows execution time taken by various tasks on basic ABC scheduling and proposed algorithm:

Table 5 Comparison of ABC scheduling & Datacenter Based ABC approach



	Datacenter Based ABC Approach				Basic ABC Approach			
No. of cloudlets	H	M	L	A	H	M	L	A
25	2.73	4.77	4.1	3.23	3.14	14.34	36.03	1.7
50	8.09	9.61	7.71	4.6	37.64	30.25	73.03	53.45
100	25.85	29.92	27.69	12.03	49.96	85.77	99.47	88.48
150	38.6	43.51	40.86	19.96	102.67	125.65	140.87	130.64
200	43.27	43.79	45.42	21.52	132.43	176.41	184.67	178.38

Table 5 shows execution time taken by various tasks using ABC approach and using proposed algorithm. In the above table, variables H, M, L, A show time taken by high, mid, low and all (high, mid and low) queues to execute all tasks.

Graph representation of above results:

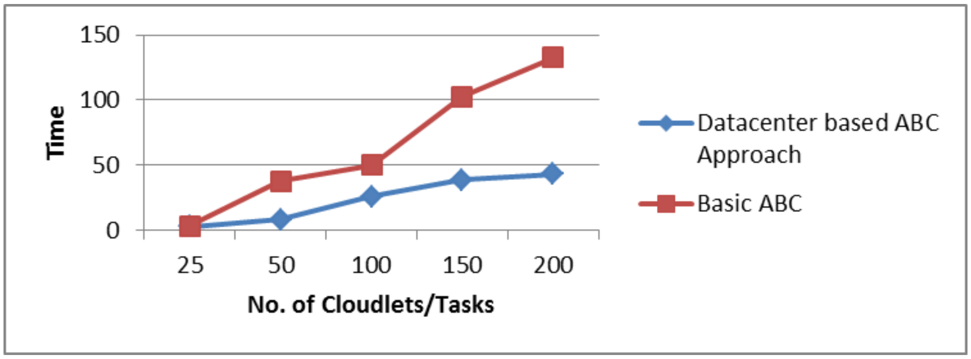


Figure 2 Execution of all tasks in High Priority Queue.

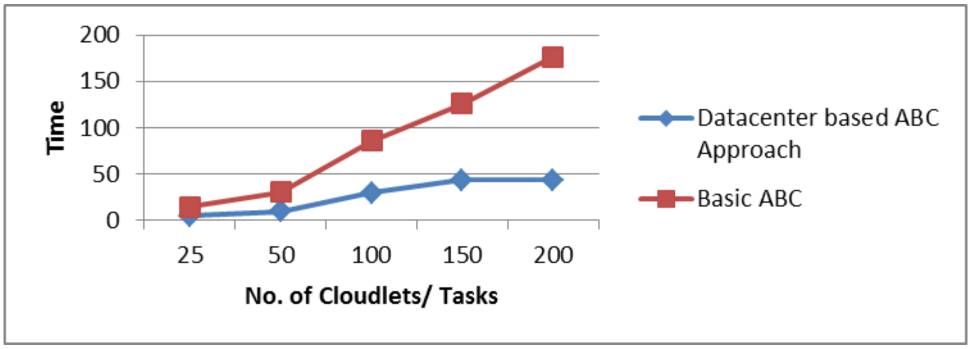


Figure 3 Execution of all tasks in Medium Priority Queue

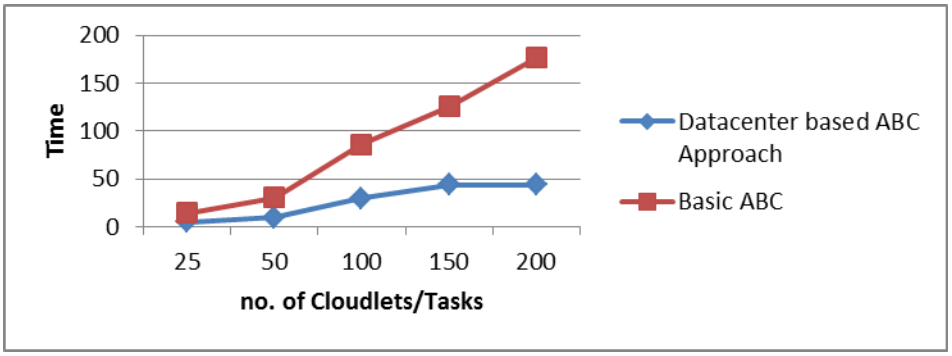


Figure 4 Execution of all the tasks in Low Priority Queue

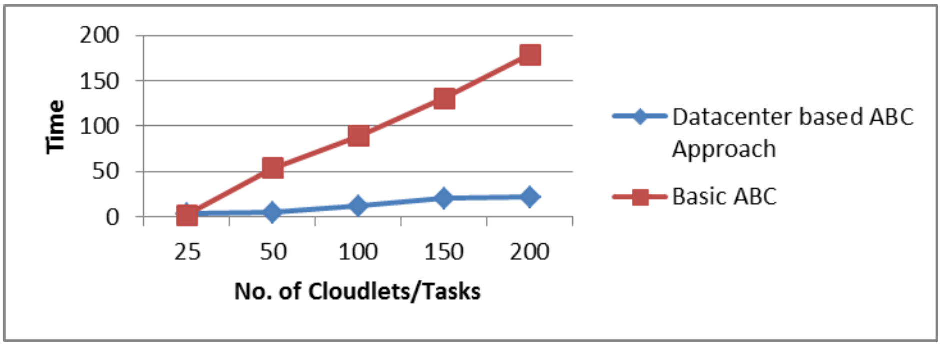


Figure 5 Execution of all the tasks in all queues

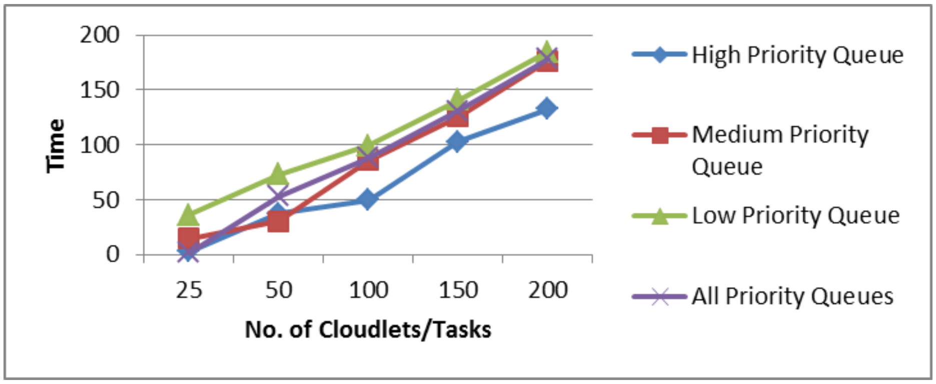


Figure 1 Basic ABC algorithm

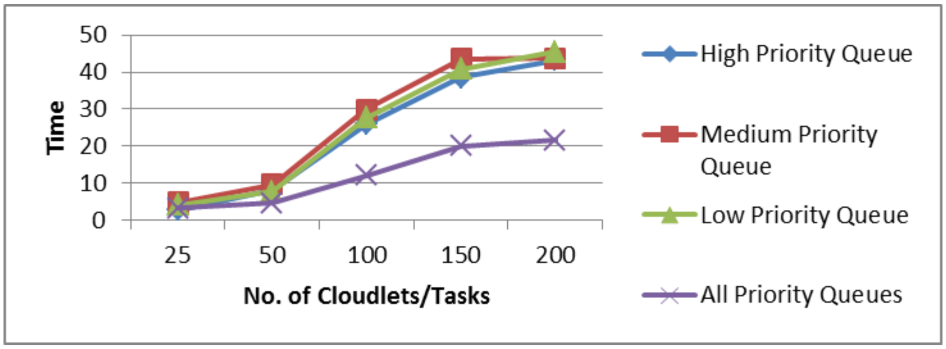
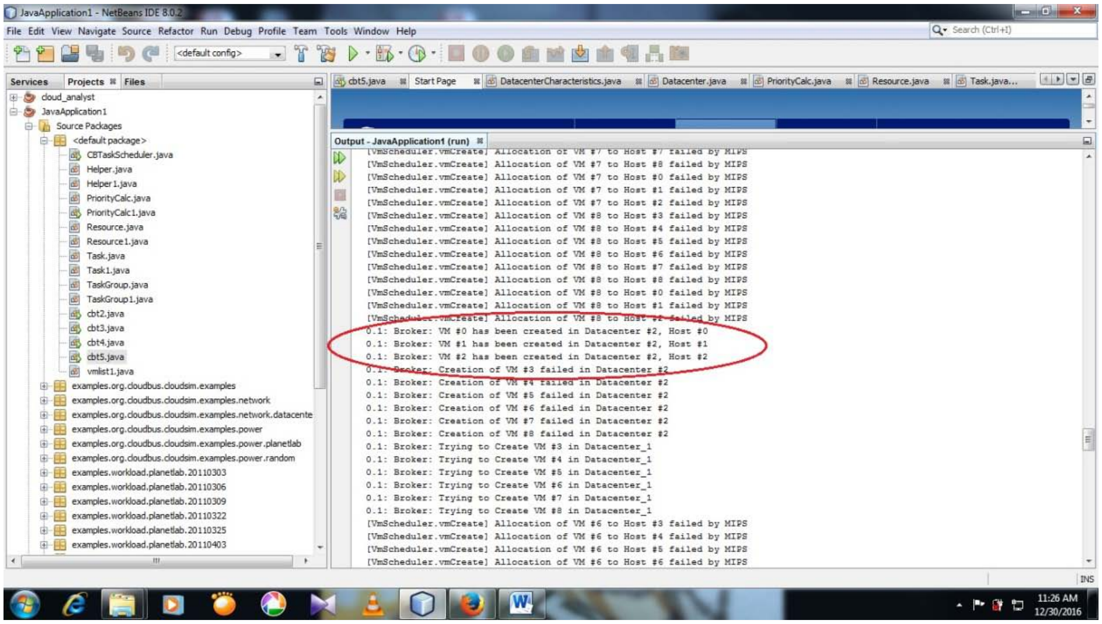


Figure 7 Datacenter Based ABC algorithm (Proposed)

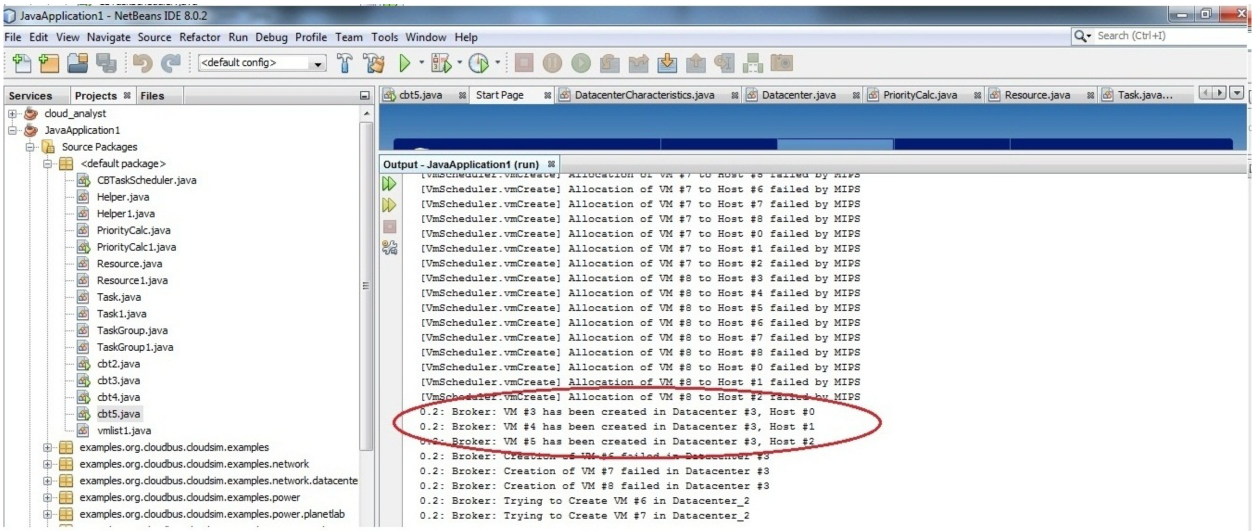
Fig. 6 shows the execution of tasks by Basic ABC algorithm and fig. 7 shows the execution of various tasks by proposed algorithms. Graphs and tables discussed above show that proposed algorithm executes tasks according to cost in less time as compared to basic ABC approach.

4.3. *Output screenshot*:


JavaApplication1 - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Services
Projects
Files
cloud_analyst
JavaApplication1
Source Packages
<default package>
CBTaskScheduler.java
Helper.java
Helper1.java
PriorityCalc.java
PriorityCalc1.java
Resource.java
Resource1.java
Task.java
Task1.java
TaskGroup.java
TaskGroup1.java
cbt2.java
cbt3.java
cbt4.java
cbt5.java
vmlist1.java
examples.org.cloudbus.cloudsim.examples
examples.org.cloudbus.cloudsim.examples.network
examples.org.cloudbus.cloudsim.examples.network.datacente
examples.org.cloudbus.cloudsim.examples.power
examples.org.cloudbus.cloudsim.examples.power.planetlab
examples.org.cloudbus.cloudsim.examples.power.random
examples.workload.planetlab.20110303
examples.workload.planetlab.20110306
examples.workload.planetlab.20110309
examples.workload.planetlab.20110322
examples.workload.planetlab.20110325
examples.workload.planetlab.20110403
cbt5.java
Start Page
DatacenterCharacteristics.java
Datacenter.java
PriorityCalc.java
Resource.java
Task.java...
Output - JavaApplication1 (run)
INS
11:26 AM
12/30/2016

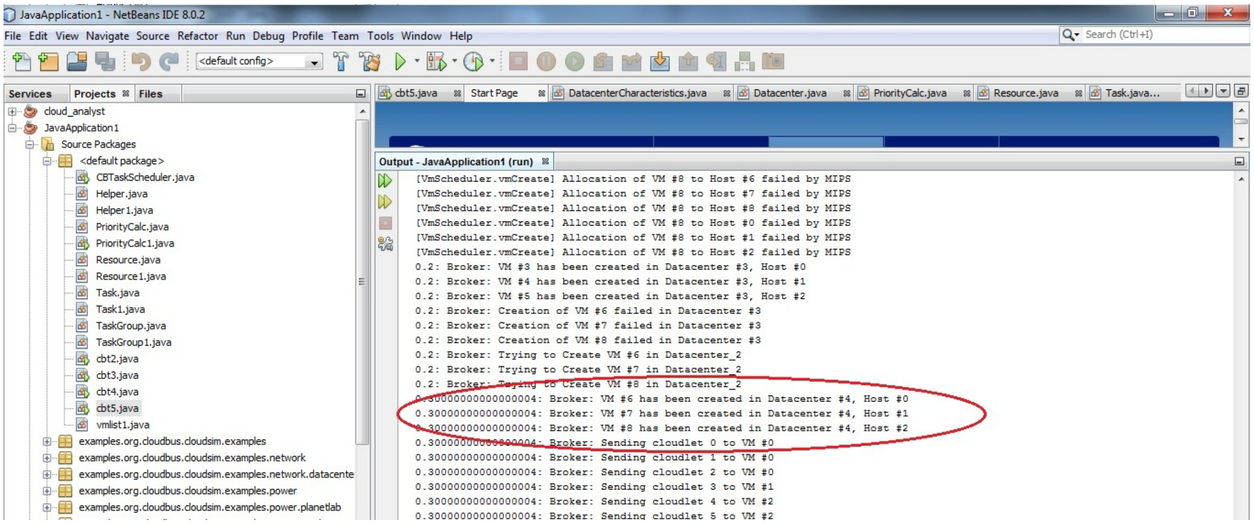
```
[VmScheduler.vmCreate] Allocation of VM #7 to Host #7 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #8 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #2 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #3 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #4 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #5 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #6 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #7 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #8 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #2 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #1
0.1: Broker: VM #2 has been created in Datacenter #2, Host #2
0.1: Broker: Creation of VM #3 failed in Datacenter #2
0.1: Broker: Creation of VM #4 failed in Datacenter #2
0.1: Broker: Creation of VM #5 failed in Datacenter #2
0.1: Broker: Creation of VM #6 failed in Datacenter #2
0.1: Broker: Creation of VM #7 failed in Datacenter #2
0.1: Broker: Creation of VM #8 failed in Datacenter #2
0.1: Broker: Trying to Create VM #3 in Datacenter_1
0.1: Broker: Trying to Create VM #4 in Datacenter_1
0.1: Broker: Trying to Create VM #5 in Datacenter_1
0.1: Broker: Trying to Create VM #6 in Datacenter_1
0.1: Broker: Trying to Create VM #7 in Datacenter_1
0.1: Broker: Trying to Create VM #8 in Datacenter_1
[VmScheduler.vmCreate] Allocation of VM #6 to Host #3 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #6 to Host #4 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #6 to Host #5 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #6 to Host #6 failed by MIPS
```

Fig.8 Creation of VMs in high-cost datacenter


JavaApplication1 - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Services Projects Files
cloud_analyst
JavaApplication1
Source Packages
<default package>
CBTaskScheduler.java
Helper.java
Helper1.java
PriorityCalc.java
PriorityCalc1.java
Resource.java
Resource1.java
Task.java
Task1.java
TaskGroup.java
TaskGroup1.java
cbt2.java
cbt3.java
cbt4.java
cbt5.java
vmlist1.java
examples.org.cloudbus.cloudsim.examples
examples.org.cloudbus.cloudsim.examples.network
examples.org.cloudbus.cloudsim.examples.network.datacente
examples.org.cloudbus.cloudsim.examples.power
examples.org.cloudbus.cloudsim.examples.power.planetlab
cbt5.java
Start Page
DatacenterCharacteristics.java
Datacenter.java
PriorityCalc.java
Resource.java
Task.java...
Output - JavaApplication1 (run)

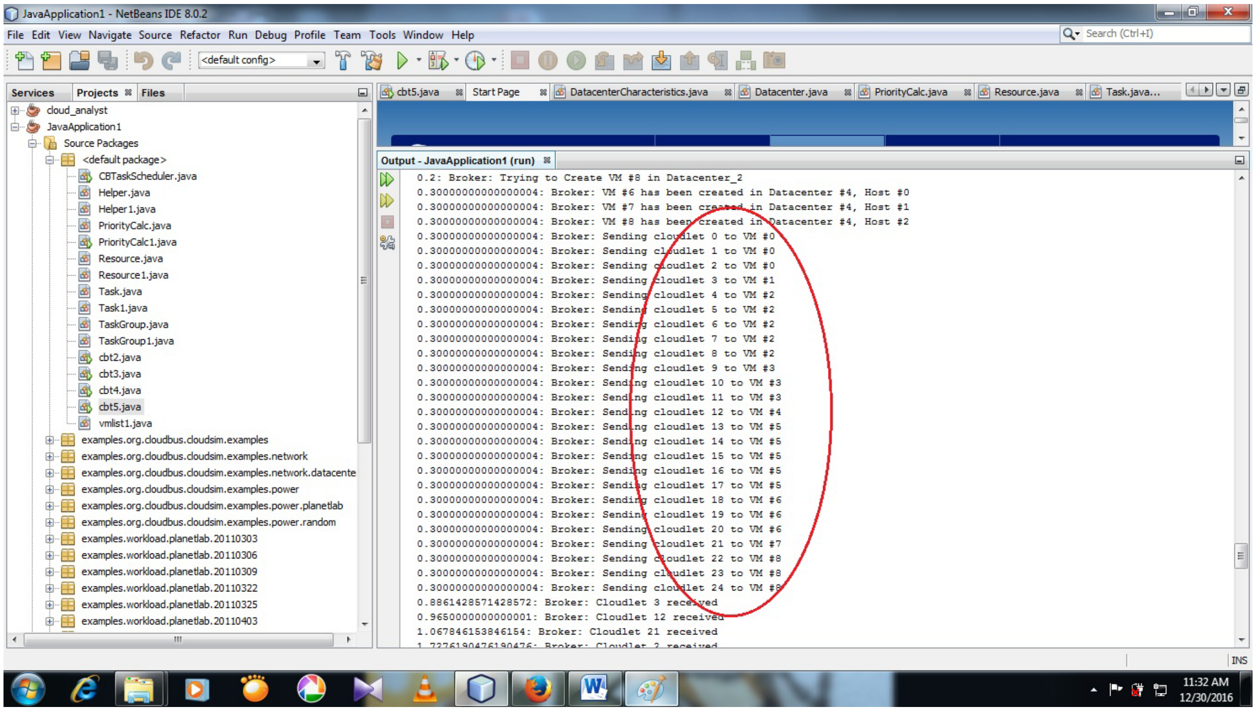
```
[VmScheduler.vmCreate] Allocation of VM #7 to Host #5 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #6 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #7 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #8 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #2 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #3 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #4 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #5 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #6 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #7 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #8 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #2 failed by MIPS
0.2: Broker: VM #3 has been created in Datacenter #3, Host #0
0.2: Broker: VM #4 has been created in Datacenter #3, Host #1
0.2: Broker: VM #5 has been created in Datacenter #3, Host #2
0.2: Broker: Creation of VM #6 failed in Datacenter #3
0.2: Broker: Creation of VM #7 failed in Datacenter #3
0.2: Broker: Creation of VM #8 failed in Datacenter #3
0.2: Broker: Trying to Create VM #6 in Datacenter_2
0.2: Broker: Trying to Create VM #7 in Datacenter_2
```

Fig.9 Creation of VMs in medium-cost datacenter


JavaApplication1 - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Services Projects Files
cloud_analyst
JavaApplication1
Source Packages
<default package>
CBTaskScheduler.java
Helper.java
Helper1.java
PriorityCalc.java
PriorityCalc1.java
Resource.java
Resource1.java
Task.java
Task1.java
TaskGroup.java
TaskGroup1.java
cbt2.java
cbt3.java
cbt4.java
cbt5.java
vmlist1.java
examples.org.cloudbus.cloudsim.examples
examples.org.cloudbus.cloudsim.examples.network
examples.org.cloudbus.cloudsim.examples.network.datacente
examples.org.cloudbus.cloudsim.examples.power
examples.org.cloudbus.cloudsim.examples.power.planetlab
cbt5.java
Start Page
DatacenterCharacteristics.java
Datacenter.java
PriorityCalc.java
Resource.java
Task.java...
Output - JavaApplication1 (run)

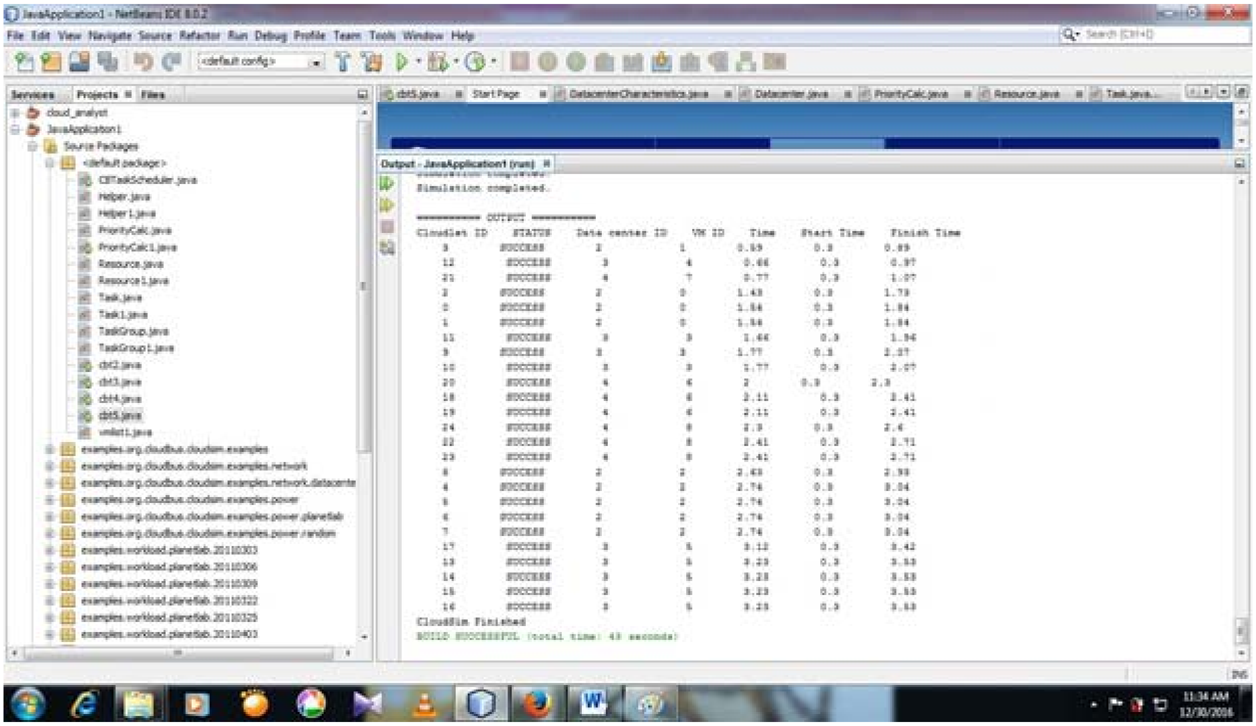
```
[VmScheduler.vmCreate] Allocation of VM #8 to Host #6 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #7 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #8 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #2 failed by MIPS
0.2: Broker: VM #3 has been created in Datacenter #3, Host #0
0.2: Broker: VM #4 has been created in Datacenter #3, Host #1
0.2: Broker: VM #5 has been created in Datacenter #3, Host #2
0.2: Broker: Creation of VM #6 failed in Datacenter #3
0.2: Broker: Creation of VM #7 failed in Datacenter #3
0.2: Broker: Creation of VM #8 failed in Datacenter #3
0.2: Broker: Trying to Create VM #6 in Datacenter_2
0.2: Broker: Trying to Create VM #7 in Datacenter_2
0.2: Broker: Trying to Create VM #8 in Datacenter_2
0.30000000000000004: Broker: VM #6 has been created in Datacenter #4, Host #0
0.30000000000000004: Broker: VM #7 has been created in Datacenter #4, Host #1
0.30000000000000004: Broker: VM #8 has been created in Datacenter #4, Host #2
0.30000000000000004: Broker: Sending cloudlet 0 to VM #0
0.30000000000000004: Broker: Sending cloudlet 1 to VM #0
0.30000000000000004: Broker: Sending cloudlet 2 to VM #0
0.30000000000000004: Broker: Sending cloudlet 3 to VM #1
0.30000000000000004: Broker: Sending cloudlet 4 to VM #2
0.30000000000000004: Broker: Sending cloudlet 5 to VM #2
```

Fig.10 Creation of VMs in low-cost datacenter



```
0.2: Broker: Trying to Create VM #8 in Datacenter_2
0.3000000000000004: Broker: VM #6 has been created in Datacenter #4, Host #0
0.3000000000000004: Broker: VM #7 has been created in Datacenter #4, Host #1
0.3000000000000004: Broker: VM #8 has been created in Datacenter #4, Host #2
0.3000000000000004: Broker: Sending cloudlet 0 to VM #0
0.3000000000000004: Broker: Sending cloudlet 1 to VM #0
0.3000000000000004: Broker: Sending cloudlet 2 to VM #0
0.3000000000000004: Broker: Sending cloudlet 3 to VM #1
0.3000000000000004: Broker: Sending cloudlet 4 to VM #2
0.3000000000000004: Broker: Sending cloudlet 5 to VM #2
0.3000000000000004: Broker: Sending cloudlet 6 to VM #2
0.3000000000000004: Broker: Sending cloudlet 7 to VM #2
0.3000000000000004: Broker: Sending cloudlet 8 to VM #2
0.3000000000000004: Broker: Sending cloudlet 9 to VM #3
0.3000000000000004: Broker: Sending cloudlet 10 to VM #3
0.3000000000000004: Broker: Sending cloudlet 11 to VM #3
0.3000000000000004: Broker: Sending cloudlet 12 to VM #4
0.3000000000000004: Broker: Sending cloudlet 13 to VM #5
0.3000000000000004: Broker: Sending cloudlet 14 to VM #5
0.3000000000000004: Broker: Sending cloudlet 15 to VM #5
0.3000000000000004: Broker: Sending cloudlet 16 to VM #5
0.3000000000000004: Broker: Sending cloudlet 17 to VM #5
0.3000000000000004: Broker: Sending cloudlet 18 to VM #6
0.3000000000000004: Broker: Sending cloudlet 19 to VM #6
0.3000000000000004: Broker: Sending cloudlet 20 to VM #6
0.3000000000000004: Broker: Sending cloudlet 21 to VM #7
0.3000000000000004: Broker: Sending cloudlet 22 to VM #8
0.3000000000000004: Broker: Sending cloudlet 23 to VM #8
0.3000000000000004: Broker: Sending cloudlet 24 to VM #8
0.8861428571428572: Broker: Cloudlet 3 received
0.9650000000000001: Broker: Cloudlet 12 received
1.067846153846154: Broker: Cloudlet 21 received
1.7276190476190476: Broker: Cloudlet 2 received
```

Fig. 11 Sending tasks or cloudlets to the particular VMs



```
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    3         SUCCESS        2               1        0.59      0.3         0.89
    12         SUCCESS        3               4        0.66      0.3         0.97
    21         SUCCESS        4               7        0.77      0.3         1.07
    2         SUCCESS        2               0        1.43      0.3         1.73
    0         SUCCESS        2               0        1.54      0.3         1.84
    1         SUCCESS        2               0        1.54      0.3         1.84
    11         SUCCESS        3               3        1.66      0.3         1.96
    9         SUCCESS        3               3        1.77      0.3         2.07
    10         SUCCESS        3               3        1.77      0.3         2.07
    20         SUCCESS        4               6        2     0.3      2.3
    18         SUCCESS        4               6        2.11      0.3         2.41
    19         SUCCESS        4               6        2.11      0.3         2.41
    24         SUCCESS        4               8        2.3      0.3         2.6
    22         SUCCESS        4               8        2.41      0.3         2.71
    23         SUCCESS        4               8        2.41      0.3         2.71
    8         SUCCESS        2               2        2.63      0.3         2.93
    4         SUCCESS        2               2        2.74      0.3         3.04
    5         SUCCESS        2               2        2.74      0.3         3.04
    6         SUCCESS        2               2        2.74      0.3         3.04
    7         SUCCESS        2               2        2.74      0.3         3.04
    17         SUCCESS        3               5        3.12      0.3         3.42
    13         SUCCESS        3               5        3.23      0.3         3.53
    14         SUCCESS        3               5        3.23      0.3         3.53
    15         SUCCESS        3               5        3.23      0.3         3.53
    16         SUCCESS        3               5        3.23      0.3         3.53
CloudSim Finished
BUILD SUCCESSFUL (total time: 43 seconds)
```

Fig.12 The result, if all tasks are executed by the low priority queue

Figures 8, 9,10,11,12 shown above show the flow of proposed approach. Fig. 8, 9, 10 shows the creation of VMs in different datacenters. Fig. 11 shows the binding to VMS to the clouds and finally fig. 12 shows the execution of tasks.

V. CONCLUSION AND FUTURE WORK

In this paper, a basic framework for scheduling in cloud computing based on cost has been proposed. Cloud was divided into three categories of datacenters based on cost. Tasks with high cost were assigned with high processing power units in the first datacenter, while the medium cost tasks were assigned to the medium processing power unit. In the last, tasks with low cost were assigned to the third datacenter with the low processing power unit. This paper calculates the expected cost of the tasks submitted and accordingly the task is sent to the datacenter. Proposed approach is based on the expected cost of tasks before their execution. The results of the proposed approach are found to be better as compared to basic ABC approach i.e. the time taken for completion of a task is less as compared to ABC. In future, the scheduling policies which will prove beneficial in terms of cost/profit to both service provider as well as client.

REFERENCES

[1] Gary Cokins, "Activity Based Cost Management", John Wiley & Sons, New York, 2001.

[2] Yair M. Babad and Bala V. Balachandran, "Cost Driver Optimization in Activity-Based Costing", *American Accounting Association*, July, 1993, pp. 563-575.

[3] Carsten Homburg, "A note on optimal cost driver selection in ABC", Management Accounting Research, 2001, pp. 197–205.

[4] Qi Cao, Zhi-Bo Wei, Wen-Mao Gong, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing", IEEE, 2009.

[5] Shachee Parikh, Richa Sinha, "Double Level Priority based Optimization Algorithm for Task Scheduling in Cloud Computing", *International Journal of Computer Applications*, January 2013.

[6] Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam, "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing", IEEE, 2010.

[7] Ashutosh Ingole, Sumit Chavan, Utkarsh Pawde, "An Optimized Algorithm for Task Scheduling based on Activity-based Costing in Cloud Computing", *2nd National Conference on Information and Communication Technology*, 2011.

[8] Zhi Yang, Changqin Yin, Yan Liu, "A Cost-based Resource Scheduling Paradigm in Cloud Computing", *12th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2011, IEEE, pp. 41.