Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

(Special Issue for ITECE 2016)

Implementation in Cloudsim of Mobile Cloud Dynamic Partitioning Technique

Prof.Chirag Vala[1], Prof.Kunal Vasani[2]

[1]Computer Science & Engineering, SLTIET
[2]Computer Science & Engineering, SLTIET

Abstract —Now a days, Mobile Users use most of Cloud Computing related services like SAAS, PAAS IAAS among them Software as a Services in that mobile user send the request to the nearer server to access some services. From that how can that requested service provides to the Mobile user is another challenge. This paper highlights the concept of how dynamic partitioning technique can improve the performance of Mobile in Cloud Computing Environment. To perform that task in real life generally mobile user doesn't know about the services of the Cloud Computing in to specific Mobile environment. In the real world Mobile user send the request through the Internet site and access that service in Cloud Environment. Then whatever the services required by user that given or set Mobile user to Cloud Environment through the Internet. That shows the Implementation in Cloudsim Environment.

Keywords: Cloud Computing, Mobile Data Offloading, Dynamic Partitioning techniques

I. INTRODUCTION

In Mobile Data Offloading in a dynamic network environment many factors are affecting like continuously changing connection status and network bandwidth. If an environment changes then causes many additional problems. For example, the transmitted data may not arrive at the destination, or the data executed on the server might be lost when it has to be returned to the sender.

First section of this report describes the Overview of the different techniques of the Dynamic Partitioning. Second section of this report explains about comparatively analysis of all the techniques. Third section Proposed Algorithm of Granularity Interaction Graph and Implementation of this technique. And in last section we conclude the different dynamic partitioning techniques.

II. PROPOSED ALGORITHM

To develop an algorithm for multi-site computation offloading in Granularity Interaction Graph that takes into account server heterogeneity, turn to an unlikely source, the computer vision community. Here I adapt the concept of Object Interaction Graph as base concept.

In this algorithm of assign Vertex and Edge weights I have consider some of the assumption and declaration of the Vertex and Edge weight taken as below.

- **GRANULARITY INTERACTION GRAPH ALGORITHM**

Vertex weight: The cost in terms of computation and/or power consumption of running a particular portion of code on the mobile device would be different from that of running it on some cloud servers, which in turn might differ among themselves in their computational capacity or cost of their availability. This is captured by the fact that each node is assigned a different cost depending on the partition of the application graph it finally ends up in or the label it is assigned.

Edge weight: The communication cost associated with movement of data and requisite messages in case parts of the application reside on different hosts varies depending on the hosts. For example, communication between two cloud-resident servers may be very fast, while communication between the mobile device and the cloud may be much slower.

The set of partitions, {p0, p1….pk}
The Set of Allocation Site {s1,s2,….sn}
Input : u is the given vertex
v is the set of vertex
Output : Graph with the Optimum Computation Cost

Step 1: Fetch the all vertex exsiting in to the site.

Step 2: Fetch the Edge weigts and Vertex weight of the site.

Step 3: Select u as partitioning vertex.
Adj[n] = Get vertices AdjTo(u)

Step 4: Make a group of vertex Array of allocation site of Partition P1.
Based on services provide by that site.

Step 5: Repeat step 1 to 4 for to make all allocation site in Partition P1.

Step 6: Visit all the vertex in to allocation site and make group of allocation site That follows appropriate Granularity level based on service.

Step 7: When one method of S1 site invoke the allocation site S2 then
Make Edge between them.

Step 8: If alias of allocation site S exsist then
Add that allocation site in to Group.

Step 9 : Repeat step 1 to 8.

Step 10: Evaluate the Graph find Optimum total vertex weight and total edge
Weight for each Partition.

Step 11 : End.

III. IMPLEMENTATION OF GRANULARITY INTERACTION GRAPH ALGORITHM

Figure shows the stepwise implementation of the algorithm.

Step 1 of algorithm assigns the Host id, Host MIPS and Host Edge weight.

Step 2 assign edge links with all node with each other according to topology.

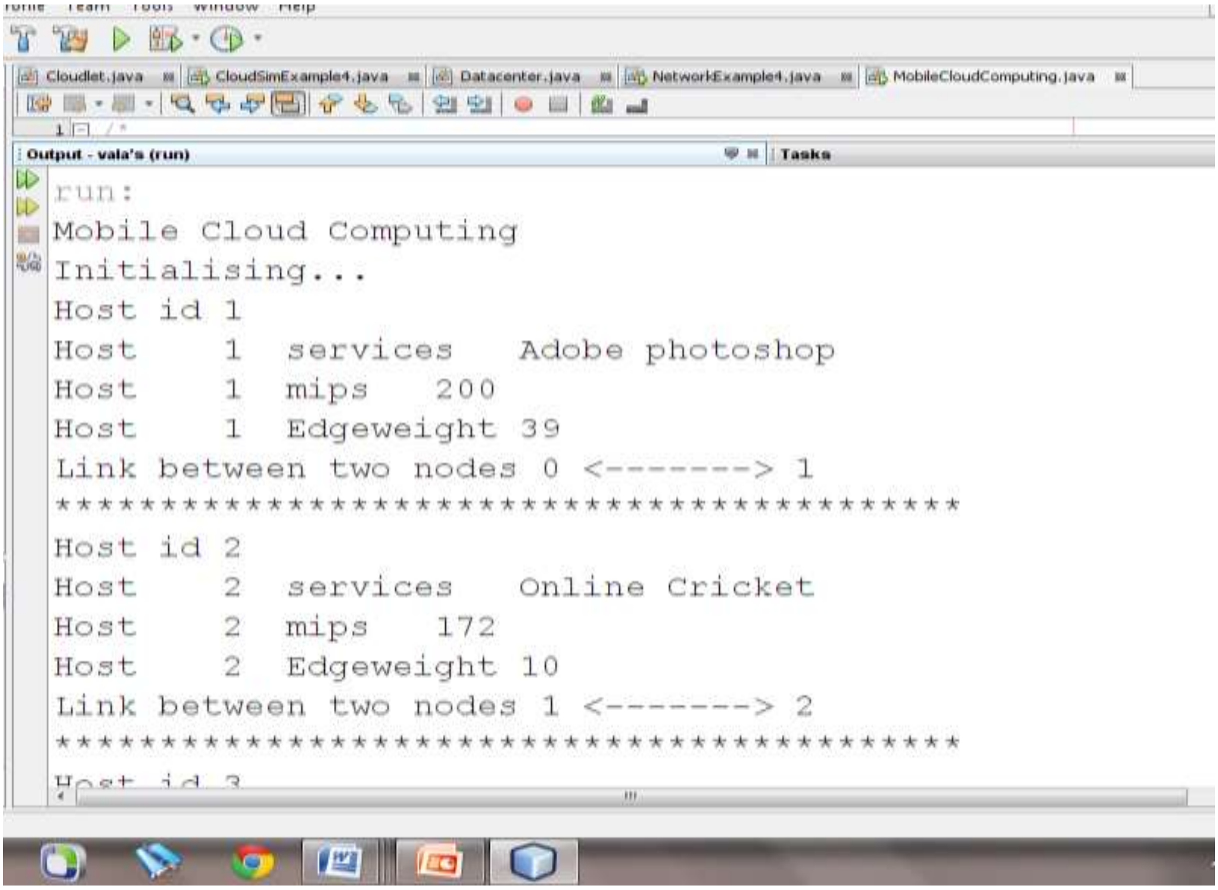


Figure: Assign Node Weight

Step 3 step shows all the Link with the all available node in Datacenter with its Host Id, Host Service, Host MIPS, Host Edge weight and Link with the node in to the Specific area.

Step 4 shows the selected service or application and its available node that having the same service. Among them it will select the highest MIPS level. If more than one Node have same MIPS than it will select minimum Edge Weight for selection a Node.

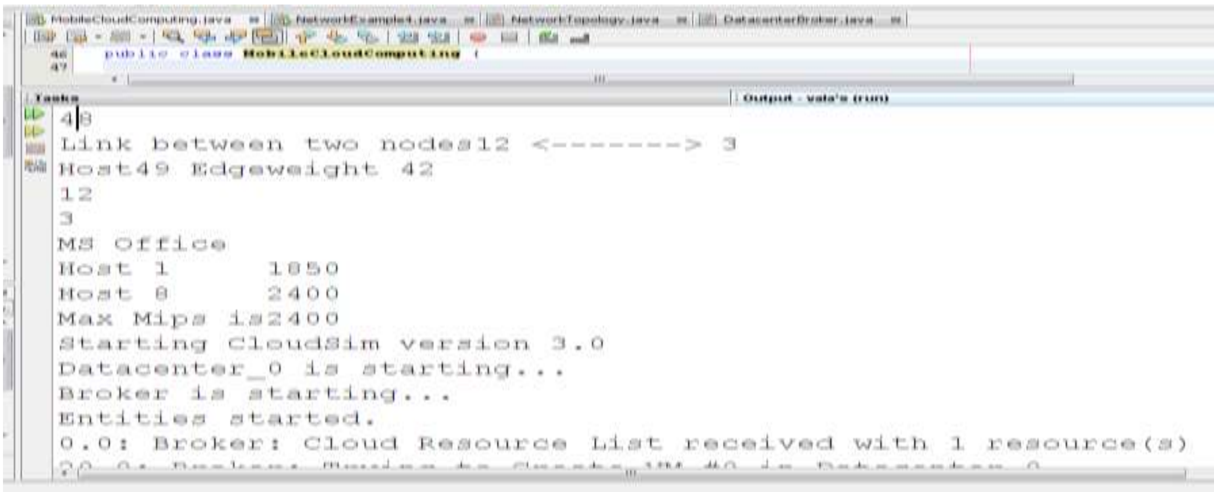
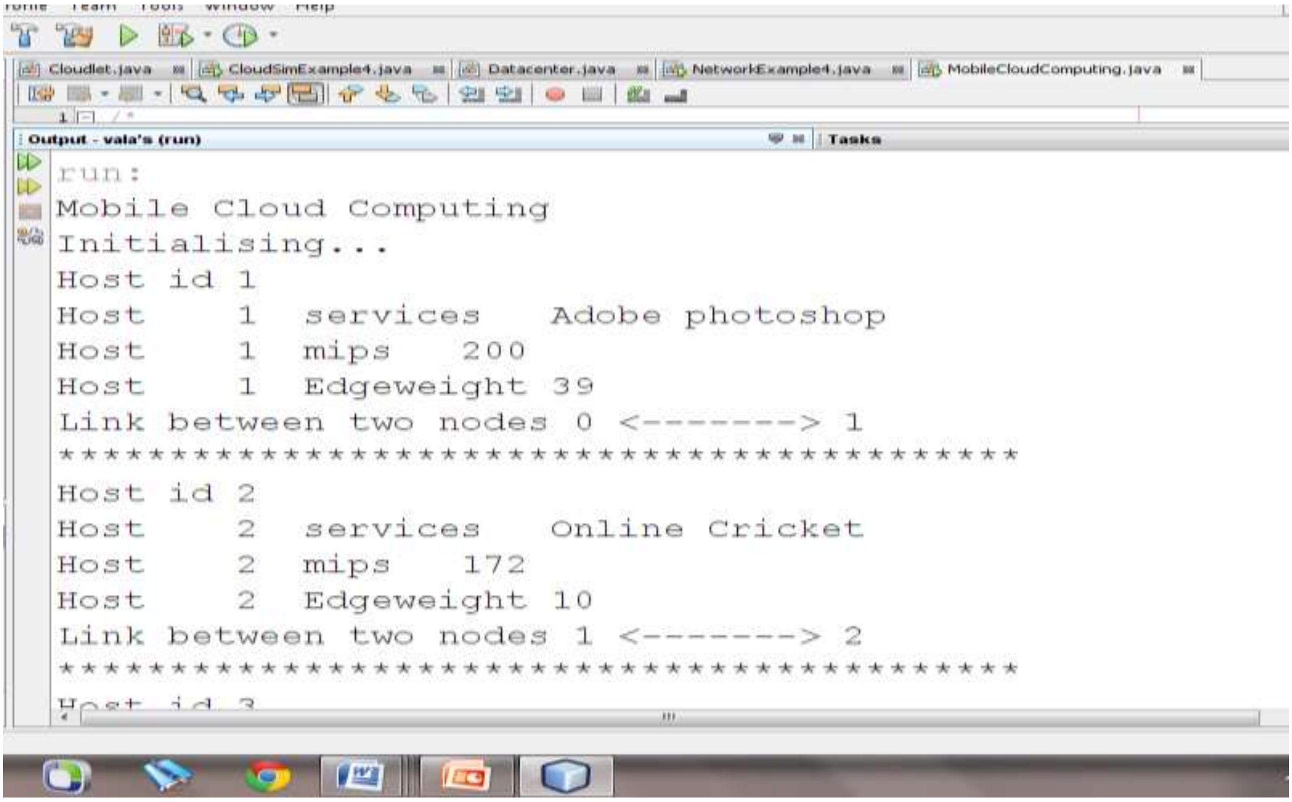


Figure 4: Selected Application and MIPS



Step 5 shows the Resultant output of the selected Cloudlet and Data Center with the Starting time, Finish time.

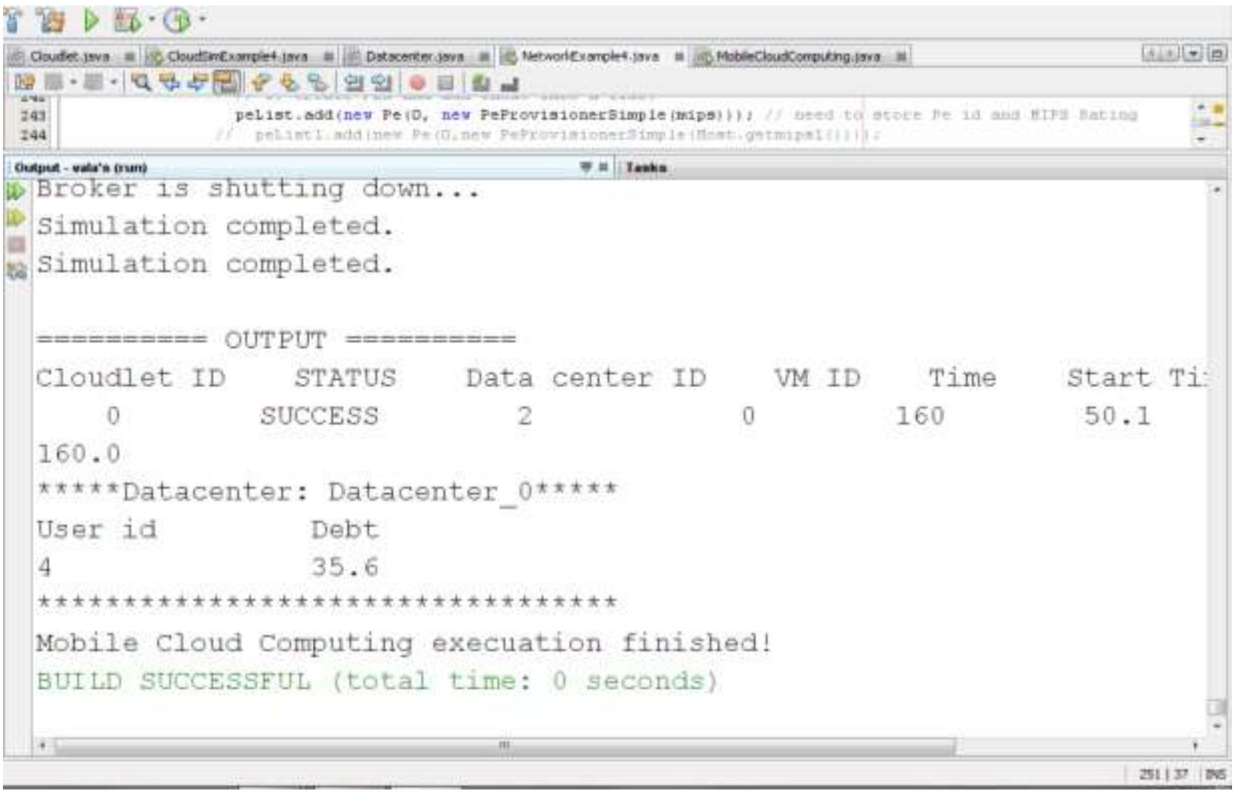
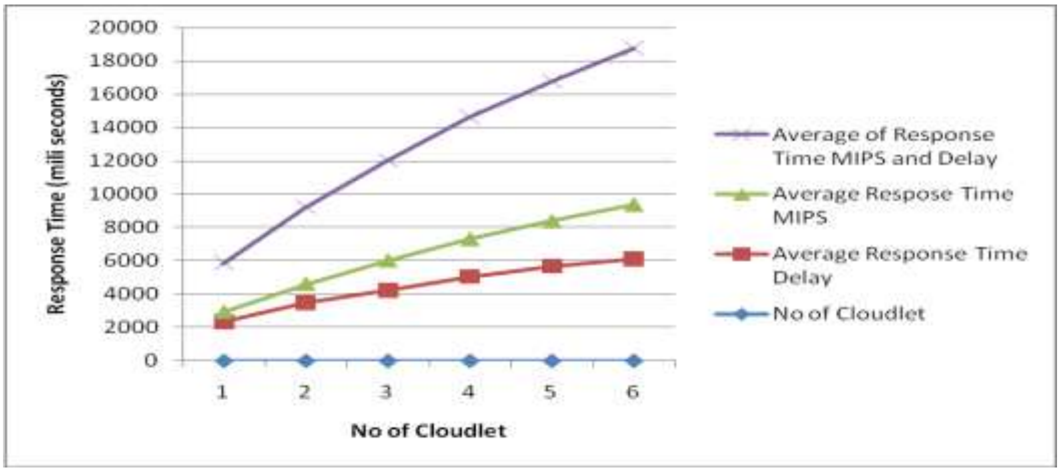


Figure 5: Resultant Output

No of Cloudlet	Average Response Time		
	Delay	MIPS	Combination
1	2345	575	2920
2	3478	1125	4603
3	4224	1780	6004
4	5050	2254	7304
5	5680	2700	8380

Table 1: Results of Proposed Algorithm

Graphs show the Response Time Vs No of Cloudlet in that it shows Average Response Time Delay, Average Response Time MIPS and Average of Response Time MIPS and Delay.

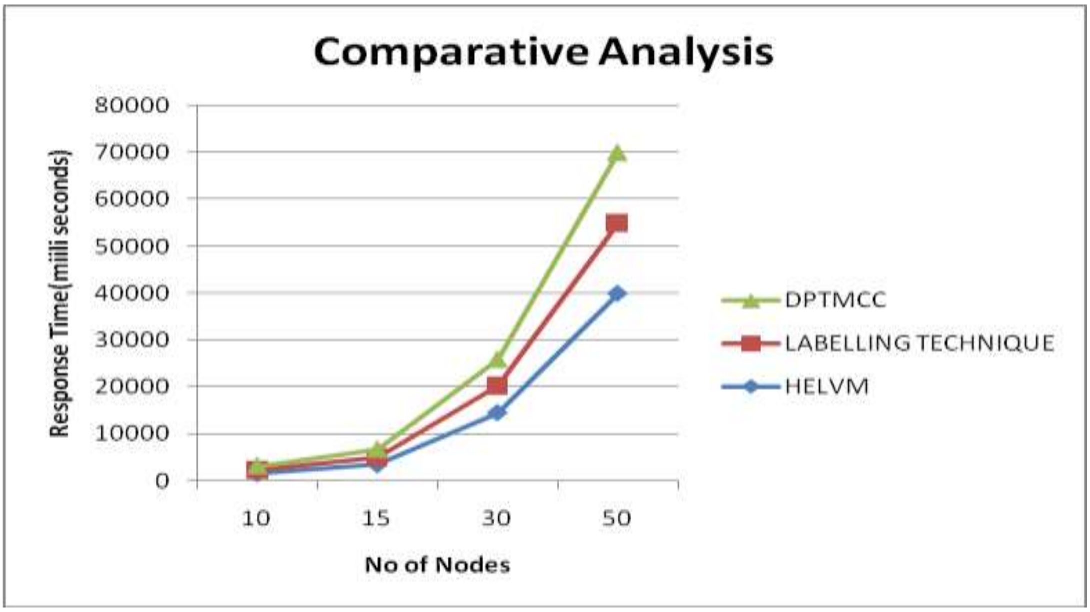


Nodes	HELVM	Labelling Technique	DPTMCC
10	1548	827	790
15	3446	1696	1550
30	14532	5762	5560
50	39907	15099	14980

Table 2: Comparative Analysis with Existing Techniques

COMPARATIVE ANALYSIS WITH EXISTING TECHNIQUES

Fig shows the comparative analysis of existing techniques and Proposed Technique DPTMCC (Dynamic Partitioning Technique in Mobile Cloud Computing).



CloudLet	DPTMCC	LABELLING TECHNIQUE	HELVM
2	5140	6120	5520
4	8340	8820	8530
6	11350	11736	11550
8	13340	13748	13670
10	15572	15940	15720

Table 3 : Max Flow Time of Three Technique

Fig shows graphical representation No of Cloudlet Vs Response time. This graph shows Flow Time of proposed technique DPTMCC, Labeling Technique and HELVM.

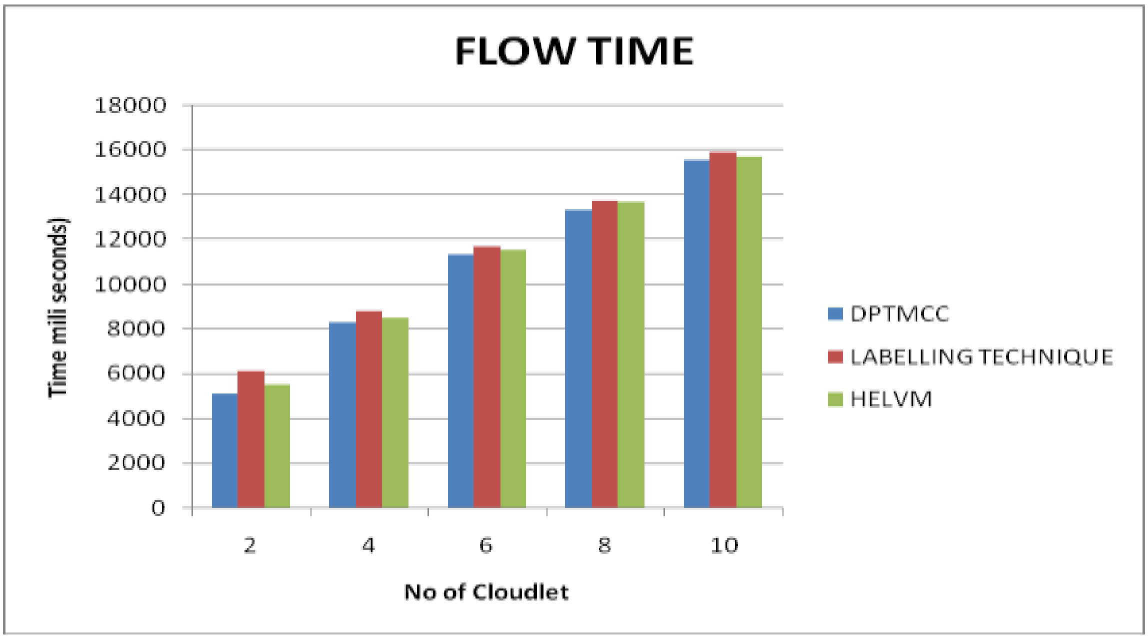
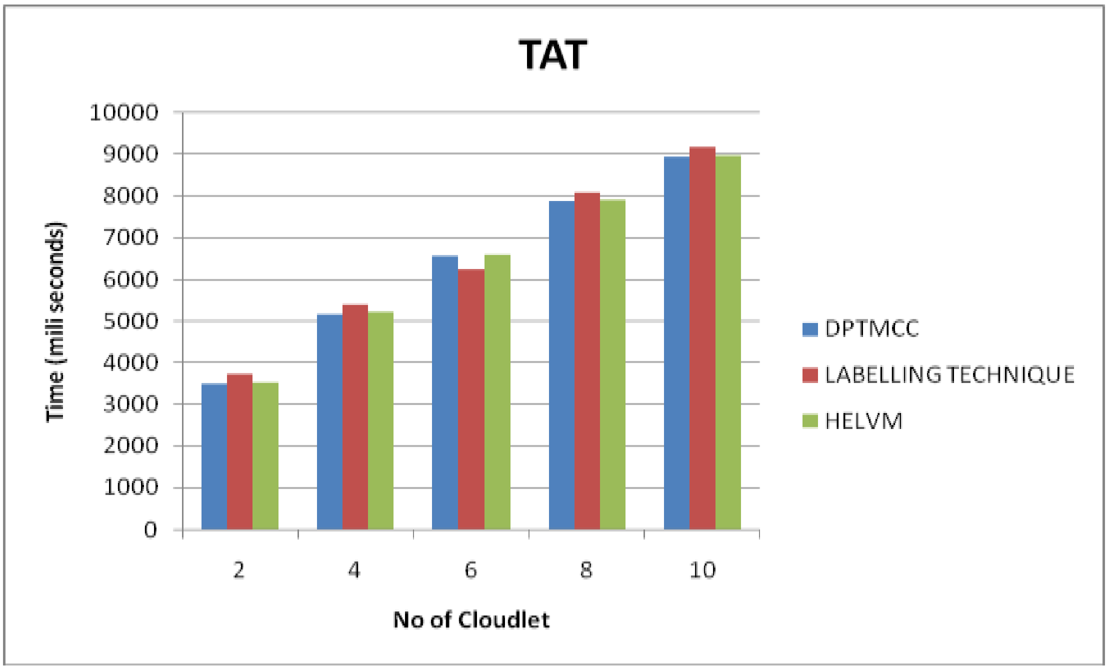


Figure 8 : Flow Time Analysis

CloudLet	DPTMCC	LABELLING TECHNIQUE	HELVM
2	3490	3720	3527
4	5173	5403	5210
6	6574	6237	6611
8	7874	8104	7911
10	8950	9180	8987

Table 5 : Turn Around Time of Three Technique

Fig shows graphical representation No of Cloudlet Vs Response time. This graph shows Turn Around Time of proposed technique DPTMCC, Labeling Technique and HELVM.



IV. CONCLUSION

Prior work in this area has focused on a constrained form of the problem: a single mobile device offloading computation to a single server. However, with the increased popularity of cloud computing and storage, it is more common for the data that an application accesses to be distributed among several servers. This thesis describes algorithmic approaches for Dynamic Partitioning Technique in Mobile Cloud Computing Environment. Using this technique it explains about how Performance of the System can increase by improving Computation and Communication Cost. This approach is based on a partitioning algorithm, and a data representation. I demonstrate that this partitioning algorithm outperforms existing technique, and that representation provides for more efficient, mobile data offloading than prior approaches.

REFERENCES

[1] Chiragkumar P. Vala, Jignesh Lakhani "Improvement of Dynamic Partitioning Technique in Mobile Data Offloading " in ASSE Journals – NOV 2013,pages 323-325.

[2] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches " in Wireless communications and Mobile Computing – Wiley - 2010,pages 212-222.

[3] Kanad Sinha and Milind Kulkarni "Techniques for fine-grained, multi-site computation offloading", 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing 2011 pages 184-194.

[4] Antonis M. Hadjiantonis, Georgios Ellinas "Converged Network and Device Management for Data Offloading", IEEE Network Operations and Management Symposium (NOMS), 2012 pages 490-493.

[5] Cheng Wang and Zhiyuan Li "Parametric Analysis for Adaptive Computation Offloading", Proceedings of the ACM SIGPLAN 2004 conference, New York, USA,2004 Pages119-130

[6] Niroshinie Fernando, Seng W. Loke and Wenny Rahayu , "Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing", 2011 Fourth IEEE International Conference on Utility and Cloud Computing pages 281-286

[7] Bo Han, Pan Hui and V.S. Anil Kumar,"Mobile Data Offloading through Opportunistic Communications and Social Participation", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 11, MAY 2012,

[8] Eemil Lagerspetz and Sasu Tarkoma ," Mobile Search and the Cloud: The Benefits of Offloading", IEEE International Conference 2011, pages 117-122.

[9] Ou S, Yang K, Liotta A, Hu L (2007) Performance analysis of offloading systems in mobile wireless environments.In: IEEE international conference on communications,pp 1821–1806.

[10] Byung-Gon Chun and Petros Maniatis, "Dynamically Partitioning Applications between Weak Devices and Clouds", MCS June 15, 2010, San Francisco, CA, USA. ACM , Article-7

[11] Shumao Ou, Kun Yang and Antonio Liotta"An Adaptive Multi- Constraint Partitioning Algorithm for Offloading in Pervasive Systems", Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06).