MPTMG:AN EFFICIENT APPLICATION OFFLOADING INTO CLOUDLETS

[1]K.Farheena, [2]B.Lalitha
[1]Student, [2]Assisstant Professor
[1]Computer Science Engineering
[1]Jawaharlal Nehru Technological University,Anantapur,India

ABSTRACT: ***Advanced distributed price-adjustment algorithm for efficient resource allocation and QoS-aware scheduling to offload user request. Here, the algorithm can converge in a finite number of iterations to balance the resource allocation at which the mobile cloud system achieves the efficiency by maximizing the total system benefit.Hence,the resultant will provide an unified service platform consisting of both cloudlets and the remote cloud by using advanced distributed price adjustment algorithm.The user may use the cloud if there are no resources available or if the user have not won the bid.So, offloading task to the cloud has become a great solution to give some resources to the users who lost the bid from cloudlets. If the user wants to offload in the cloud he may request the cloud instead of accesing the cloudlet resources.***

KEYWORDS– Offloading, Cloud computing, Resource bidding, Cloudlet.

1.INTRODUCTION

Mobile cloud computing system is an emerging mobile computing paradigm, has received a lot of recent interests [1].Cloud computing provides the better solution in providing the resources. Cloud computing provide services based on pay-as-you-use principle and provides a model for enabling ubiquitous, convenient on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction, as NIST[1] describes.

Mobile cloud computing is an extension to the unmodified cloud to a second level consisting of self-managed data centers with no hard state called cloudlets.The term mobile cloud is generally referred in two perspectives: (a) infrastructure based, and (b) ad-hoc mobile cloud. In infrastructure based mobile cloud, the hardware infrastructure remains static. Alternatively, ad-hoc mobile cloud refers to a group of mobile devices that acts as a cloud and provides access to local or Internet based cloud services to other mobile devices.

The major characteristics of mobile cloud computing are: a) break through the terminal hardware limitations, b)convenient data access, c)intelligent load balancing, d) effectiveness of task processing, e) on-demand service to reduce costs, and f) elimination of regional restrictions.

2.RELATED WORK

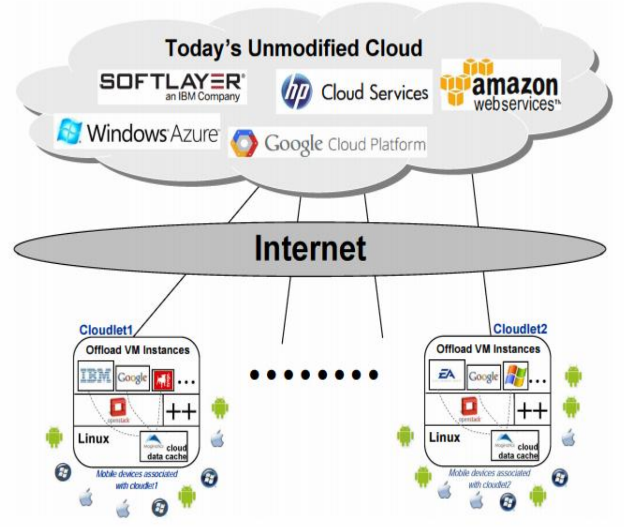
a.Cloudlet and cloud Offloading with Resource Provisioning

The authors in [2] proposed a technique to schedule the computation offloading with an available number of cloudlet resource elements and partitioning of computation tasks between the mobile side and the cloud side to minimize the average delay in processing the application. To minimize the computation time of an application, a heuristic algorithm [4] is used to offload tasks to the cloudlet such that the parallelism between the user and the cloudlet is maximized.Here there exists multiple cloudlet to provide resources to the users. Those resources are allocated by cloudlet based on bidding requests from multiple users. The users who won the bid in accesing the resources allotted by cloudlets with higher bidding cost will maximize the utility value of cloudlets. Then if the user lost the bid, then user can offload in the cloud only if cloud accepts the user request.

b.Two side Market Game

To study the matching process, Gale and Shapley [5] introduced two side market game models which have two different player groups. A player from one group of the market should be matched with a player from other group. In the market game [7], each player in the game has their own preference to select the player from other side,here competition exist among group members i.e., which group member should win the game. The same two side market game we use here,in which users can select their own cloudlets which are suitable for their request. In the proposed work there will be a competition among users to win the cloudlet for access the resources considering the bidding price[8]. Then we considering both bidding price and time to reduce the competition complexity. So the user who bids with the best price will win the bid.

c.System Model



Fig(1): Two Level Architecture of Cloud-Cloudlet

Generally, cloudlet is a resource-rich server or server cluster and it is well connected to mobile devices via a high-speed local area network (LAN). Cloudlet can be viewed as a "data center in a box" whose goal is to "bring the cloud closer" [3]. Cloud is a large space where large amount of data can be stored by many users, if data is uploaded in cloud then,we can access the data anywhere.In fig(1)users will seek resources from the remote cloud in case1. if there are no cloudlets nearby and case2. when the local cloudlets do not have space or resources to support the service of a user, If the nearby cloudlets are not available then user can offload directly in cloud.

d.Algorithm-AdvancedDistributed Price-Adjustment Algorithm

```
1. Initialization (For every user VM request)
Case1.(if user wants to directly offload his request to cloud)
    a. User requests cloud to provide resources.
    b. If cloud accepts then user may offload his request .
Case2.
2. The user broadcasts its VM request to the nearby cloudlets,cloudlets will respond with their basic charges, then the user sets its initial bidding price for all nearby cloudlets.
3. Compute its waiting time according to its arriving time.
4. Calculate the utility values (i.e.,utility value is the difference between the resource minimum cost and the user bid value) based on the bidding price and its waiting time.
5. Identify the cloudlet that makes the VM's utility value larger than 0 as the VM's available cloudlet.
6. Select its favorite cloudlet from the set of available cloudlets according to the utility values, and send its offloading request to this cloudlet.
while user's offloading request or cloudlet's reject is issued, do
7. Cloudlet side: Cloudlet Selection (For cloudlet which receives the offloading requests from users)
8. Select its favorite VMs according to its utility value and under the constraint of its available resources.
9. other than the favourite selected favorite VMs all are rejected.
10. User side: User Requesting Update (For user whose offloading request is rejected by one cloudlet)
11. if VMi's request is rejected by cloudletj , then
12. VMi must raise the bid price.
 else
13. For other cloudletj, VMi's bidding price remain the same
 end if
14. Calculate the utility values and identify the set of its available cloudlets.
15. if no available cloudlet exists, then
16. No offloading request is sent.
else
17. Select its favorite cloudlet from the set according to the utility values, and send cloudlets offloading request with new bid price to the selected cloudlet.
end if
18. user may choose the cloud directly to offload resources.
19. If user lost the bid then cloud will accept user request
end while
```

We proposed the above algorithm,advanced distributed price adjustment. In this, initially user's sends request to cloudlets, then the available cloudlets will display with their resources, minimum cost and other details. Now users will choose his favorite cloudlet and they send bidding request to the cloudlet with their best prices. Now particular cloudlet will check the bidding requests, which are bided by the users and cloudlet will accept the bidding request of highest bidding price users.Always user bidding price request should be more than the minimum cost given by the cloudlet. Now the winner of the bid can offload the data in to the cloudlet and they can download the data [6].

The advanced distributed price adjustment algorithm to improve the reliability of VM's along with cloud and cloudlets.

e.Basic Utility Function

Cloudlet can check the utility gained based on the resources available user bid price. In order to maximize the utility value the cloudlet accepts the highest bid values [6]. On the cloudlet side, as cloudlets are selfish, they tend to serve the users that can maximize their benefits. The lowest price a *cloudlet*j would sell its resources to *VM*i is represented as σij.where, $\vec{\sigma}$ij ∈ *Rd* and *d* is the total number of resource types in the cloudlet. *S*j denotes a the set of VM's selected to run in *cloudlet*j **,** the net benefit of *cloudlet*j **c**an be expressed as,

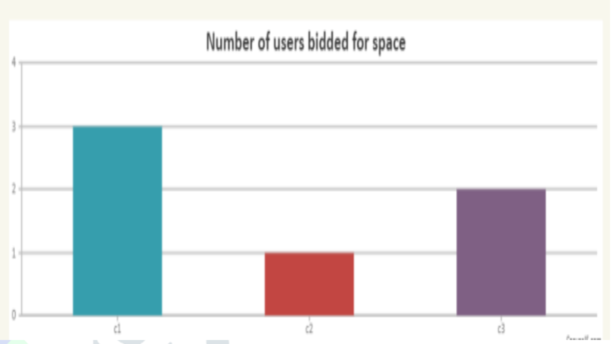
$$u_j(S_j,\vec{p}_j) = \sum_{VM_i \in S_j} \left(\sum_{1 \le k \le d} \vec{p}_{ij}[k] \cdot \vec{v}_i[k] - \sum_{1 \le k \le d} \vec{\sigma}_{ij}[k] \cdot \vec{v}_i[k] \right)$$

The utility function of a VM is related to its net benefit, which depends on the resources provided and the price charged by a selected cloudlet. The benefit of a *VM* increases as it obtains more resources, and decreases if the resource price is higher.

3.EXPERIMENTAL RESULTS

A.Number of bids:

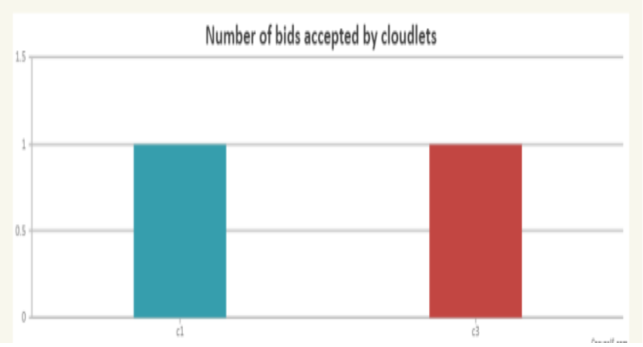
By considering the users bid count for the space (resource), for a particular cloudlet is shown in this fig(2). In fig(2) x-axis refers to number of cloudlets, in y-axis refers to users bid count.



Fig(2): number of users bade for space

B.Accepted user's bids by cloudlet:

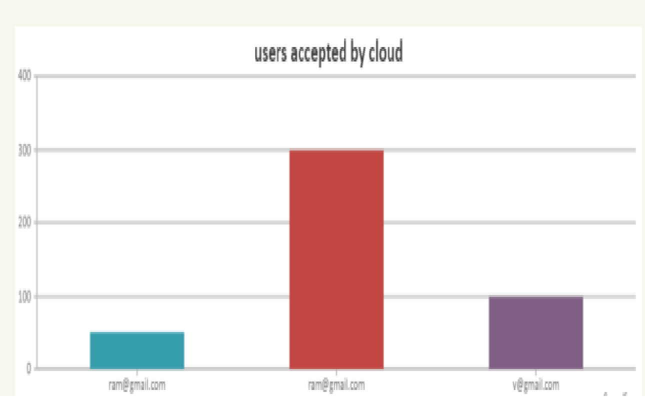
The users accepted bid count is considered in this figure. In fig(3) the users who bid more than minimum cost are the winners of the bid, so that user count is taken to evaluate the results. The x-axis refers to number of cloudlets and the y-axis refers to number of accepted bid count.



Fig(3): number of bids accepted by cloudlets

C.Accepted users by cloud:

Here the users who lost the bid and the users who chooses the cloud for offloading are considered to evaluate the results. The x-axis refers the users list and the y-axis refers the storage(bytes) requested by user in fig(4).



Fig(4): number of users accepted by cloud

4.CONCLUSION

By using advanced distributed price algorithm, user offloads data in cloudlet and cloud based on bidding process. Here, we are providing a unified service platform consisting of both cloudlets and the cloud to improve the reliability in providing resources. When the cloudlets do not have enough resources to serve all the users, then user offloads in cloud, and the bid rejected users also offloads in cloud.

REFERENCES

[1] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani, “An open ecosystem for mobile-cloud convergence,” IEEE Communications Magazine, vol. 53, no. 3, pp. 63–70, 2015.

[2] L. Yang, J. Cao, H. Cheng, and Y. Ji, “Multi-user computation partitioning for latency sensitive mobile cloud applications,” IEEE Transactions on Computers, vol. PP, no. 99, pp. 1–14, 2014.

[3] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, “Cloudlets: at the leading edge of mobile-cloud convergence,” in MobiCASE, IEEE, 2014.

[4] M. Jia, J. Cao, and L. Yang, “Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing,” in Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, pp. 352–357, IEEE, 2014.

[5] D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” The American Mathematical Monthly, vol. 69, no. 1, pp. 9–15, 1962.

[6] Distributed Multi-dimensional Pricing for Efficient Application Offloading in Mobile Cloud Computing Kun Xie1,2, IEEE, member, Xin Wang2, IEEE, member, Gaogang Xie3, IEEE, member Dongliang Xie2, IEEE, member, Jiannong Cao4, IEEE, fellow, Yuqin Ji2, Jigang Wen3.

[7] B.Lalitha “Evolution of Two sided Markets in game theoretic pricing”, ISSN 2278-0947 ,Vol 3(1),Jul-Dec 2013

[8] T. Eisenmann, G. Parker, and M. W. Van Alstyne, “Strategies for two-sided markets,” Harvard business review, vol. 84, no. 10, p. 92, 2006.