IEEE Sponsored 2nd International Conference on Innovations in Information,Embedded and Communication Systems
ICIIECS'15

Allocation and Sharing Infrastructure Resources via Social Networks using Preference based on Trust Method

Hima K S
Dept. of Computer Science and Engineering
Nehru College of Engineering and Research Centre
Thrissur, India
himaks36@gmail.com

Avanish Kumar Singh
Dept. of Computer Science and Engineering
Nehru College of Engineering and Research Centre
Thrissur, India
aavi13@gmail.com

***Abstract*— With the increasingly ubiquitous nature of Social networks and Cloud computing, users are exploring new ways to interact with, and exploit these paradigms. Social networks are used to reflect real world relationships that allow users to share information and form connections between one another. Leveraging the pre-established trust formed through friend relationships within a Social network a dynamic cloud-like platform is formed where the provisioning of Cloud infrastructure occurs through "friend" relationship. As users may have complex preference structures concerning with whom they do or do not wish to share their resources, resources can be effectively allocated within a social community offering resources on a best effort basis. Genetic algorithm along with Threshold Acceptance algorithm is used and performs best for large datasets. Preference can be mentioned manually or by using automatic deduction methods.**

***Index Terms*—preference, social networks, cloud, allocation.**

I. INTRODUCTION

Cloud computing, often referred to as simply "the cloud," is the delivery of on-demand computing resources. Today cloud computing covers anything that involves delivering hosted services over the internet. The services are generally divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Cloud computing has real benefits when it comes to data sharing and its these advantages which has led to its adaption in multiple organizations at a fairly rapid pace. In spite of these benefits the main concern regarding cloud services today are the trust and accountability. These concerns arise between the consumers and the providers. Now most of the cloud service providers suggest different methods to address these problems. Some methods can be used such as SLA's to provide trust.

This paper solves the issue of trust and accountability by bringing in a new concept called "Social Compute Cloud" [1]. A social cloud is a dynamic cloud-like sharing platform utilizing the concept of social network. A social network is a digitally encoded form of socially connected peers which have social relationships among its members. The problem of trust can be solved by including social network into this dynamic platform. Social networking sites such as Facebook, twitter, Google+ etc. plays an important role in everyday life. Here social networking is used as a medium for sharing and allocating resources because of its widespread use and size. Social networks are chosen as a platform for sharing due to its vast importance today. As per Pew Internet Project's research related to social networking, from the year 2005 to 2014 there is an increase from 8 percent to 74 percent regarding the percentage of adults who use social networking sites while accessing the internet. Also as of September 2014, 71% of online adults use Facebook.

The concept of social cloud is motivated by the vision that the socially connected peers could access the resources which they are in need of but which they are not in possession of. So the concept of social compute cloud brings the idea that the resources are shared using a social network platform.

. The main thing to be noted here is that these resources are only shared and there are no monetary benefits associated with it. In the earlier papers of Social cloud [2] there was a virtual credit model, but this paper of Social Compute Cloud [1] does not tell about any of the selling resources but only sharing of resources through a Social Clearing House. The main benefit associated with this model is that the users can have preferences with whom they can share their resources. To support user preferences several algorithms are considered for the matching of users i.e. the consumers and providers.

II. RELATED WORK

Today the importance of social networks has increased. So people are using these social networks for different types of collaborations like scientific portals. There is Social Volunteer Computing[3], proposed by McMahon and Milenkovic, which is used for the peer to peer storage, and here also there is an underlying social relationship between the customers and the providers, but unlike Social Compute Cloud there is no bilateral exchange of information in this.

There is an also off-site storage system like FriendBox [4], a friend to friend system which provides secure and private

storage system. Cooperative backup systems like FriendStore [5] is proposed where the users are given choice to what nodes are required for them for the backup.

There are several algorithms that are considered in the preference matching. C. Haas, S. Caton, and C. Weinhardt [6] proposed several algorithms and various incentives for the social cloud. Here the authors investigate how the resources are allocated effectively and also the performance of various algorithms that they have considered. The authors also considered that the Social Compute Cloud can be implemented by extending Seattle [7]. Seattle is an open source peer-peer educational platform which can be used to implement virtualization using its lightweight virtualization layer.

III. System Model

A social compute cloud is provided to enable access to resources which are contributed by the socially connected peers. Here the resources are owned and consumed by the members of the social community. By this the consumers are able to access the resources that are contributed by the providers. Through this the consumers can execute applications on compute resources that are provided by the providers.

The architecture of social compute cloud is given in Fig. 1. There are three main sections in the architecture. They are the Social Cloud Platform, Technical adapter and the Socio economic model.

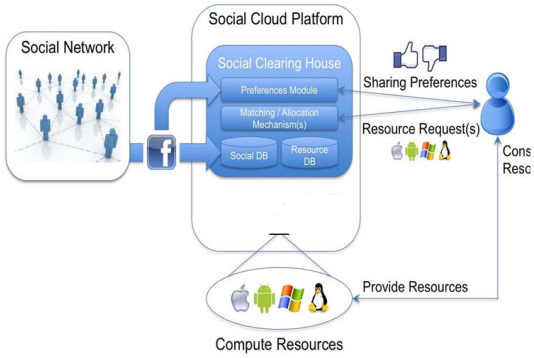

Fig. 1 Social Compute Cloud Model

A. Social Compute Cloud Platform

There should be a platform which should coordinate the basic functionality of the system. The Social Cloud Platform is the main part of the architecture. It contains the social clearing house, the middleware, the databases and the matching mechanisms.

The **Social Clearing House** is the central part of the system. It contains mainly two databases: one for maintaining the details regarding the resources and the other which contains details regarding the users. The details regarding the users can mean the social ties and other preference details. It also contains the details regarding with whom the users can share data.

There should be a **middleware** in order to implement virtualization. It should provide various sandboxing mechanisms. Seattle is mentioned in the model as a middleware to access the resources. The lightweight virtualization can be provided easily using the Seattle platform. Actually Seattle is an open peer to peer virtualization platform.

A **socio technical adapter** provides access to the users' social graph. Through this the social ties or the graph of a particular user can be determined. Once the social ties of the user are being interpreted the social clearing house checks for the preferences associated with a particular user. A preference module is used for this.

The **matching mechanisms** provide the necessary allocation of resources for the customers.

B. Technical Adapter

To enable sharing of resources there should be access to users' social network. For this purpose a technical adapter is used. Here Facebook is considered and as an access to users' profile, an authentication using the API's are considered such as Facebook Connect. Once the social graph is obtained the question is how to interpret the social ties. The social tie interpretation include how to provide and with whom to share the resources. There are many methods and any one can be adopted. It can be by 1) manually ranking the friends numerically, 2) adopt some deduction methods to calculate the preference or 3) decide using some mathematical models.

There are many advantages and disadvantages for these methods. The method mentioned initially is very easy and can be done manually. But the problem arises when the social graph is very large. In such cases it will be difficult to rank each friend manually. The other methods will be convenient for large data sets. But such methods will have more computational overhead.

Another important concept to be considered is the type of preference that exists. The preference profile consists of the ranks that a user has towards another user. So in such a case there can be many choices. Preference is represented through ranked ordered lists. A preference profile for user A is said to be complete if preference ranking is given for every other users. It is incomplete if user A leaves at least a single another user unranked.

C. Socio Economic Model

This part of the model shows the allocation process of resources. Once the preferences are captured the request for the resources and the availability of resources is to be checked. All these are done in the Social Clearing House i.e. the central part of the system. The overhead associated with this model is the storage of the necessary information associated with it. When the supply and demand are captured certain constraints are specified such are stability, fairness and the allocation process is done according to that. This model concentrates only on non-monetary benefits. The requests are being captured and the allocation is done using certain algorithms which is discussed in the next section. The allocation is done on one-one basis and this paper does not focus on many to many or one to many basis.

IV. PREFERENCE RANKING

Preferences can be mentioned manually or can be captured using the trust method. Several algorithms are used for the preference based matching. The method for preference capturing manually i.e. each users give numerical preference to every other user is very simple. This method is easy and simple. In the case of relatively small number of users it is really efficient method, but when the number of users or when the number of friends for each user exceeds and reaches for almost 500 or 1000 manual ranking of friends will become really difficult. So there should be other method to provide the preferences automatically. So the trust method is proposed. In the real world scenario the preference may change over time. So there will be difficulty in ranking people with whom we have not interacted so far. So there is a need for trust networks. These trust networks incorporate the feedback of other users in addition to own experiences about transactions with other users. If users did not interact with certain other users before, it might be difficult to obtain a preference ranking due to the lack of knowledge about these other users. By using trust networks and the corresponding trust values, users might be able to improve the accuracy of their preference rankings over time.

V. PREFERENCE MATCHING ALGORITHMS

There are several algorithms that give stable matches based on certain preference structures [6]. Depending on the quality guarantees that an algorithm offers, they can be distinguished into exact, approximate and heuristic algorithms. Exact algorithms yield the optimal solution for a given scenario, and approximate algorithms guarantee that the solution quality is within a certain bound of the optimal solution. Heuristic algorithms, in general, do not provide such a quality bound, yet have other advantages such as the flexibility to consider multiple simultaneous objectives.

A. Deferred Acceptance Algorithm

Gale and Shapley introduced this algorithm. This algorithm works for preferences that are without ties. So if there exist a tie then the tie has to be broken before giving as the actual input. If there is an unmatched user of side 1 who has not proposed to all acceptable users in their preference list it is proposed to the partner that is highest rank in the list. If the partner is currently unmatched then that pair is matched and allocated. Also if the proposed partner is currently matched to a lower rank then it is temporarily detached and then matched to the side 1 partner. It can be easily adapted to cope with incomplete preferences, and indifferences are usually handled by breaking the ties first before applying the DA in its standard formulation. The disadvantage of this algorithm is that in case of incomplete preferences the DA does not guarantee a matching of maximum size. DA runs in $O(n^2)$ where n is the size of a (symmetric) market side.

B. Welfare Optimal Algorithm

For the problem of finding a welfare-optimal stable solution in case of strict and complete preferences Welfare-Optimal algorithm that yields a solution in polynomial time is used. If n is the size of one side in a symmetric setting, the runtime of the algorithm is $O(n^4)$. The algorithm uses the concept of rotations. Similar to Deferred Acceptance algorithm it can be used with strict preferences or with indifferences. The ties should be broken initially. However, in case of indifferences WO only guarantees the welfare-optimal stable solution for a given tie breaking; the global welfare-optimal matching cannot be guaranteed.

C. Genetic Algorithm with Threshold Acceptance

GAs start with a (usually randomly created) initial set of potential solutions (the population) and evolve this solution set by applying certain mathematical operations on them. The quality of a potential solution is determined by its fitness, i.e., how well it performs with respect to the given objective function. For the matching problem, each chromosome represents a solution. A chromosome consists of several genes, where each gene encodes a provider-requester match of the solution. In other words, when a solution has m matches, the chromosome has m genes, and each gene consists of two identifiers, one for the provider, and one for the requester. As fitness functions for two-sided matching, the maximization of stability, welfare, fairness, number of pairs, or a combination thereof are used. As each chromosome only encodes a set of matched pairs, the preferences of the users are needed to evaluate its fitness.

Threshold acceptance algorithms try to improve a given solution and hence are suitable for finding (especially local) improvements. Compared to GAs which work on a population of different solutions in the solution space, the performance of a TA depends on the quality of the starting solution. TA is an example of local search heuristics, where a given starting solution is improved by sequentially adjusting the solution and accepting adjustments within a certain threshold.

So a new algorithm GATA is formed combining the benefits of both Genetic Algorithm and Threshold Acceptance Algorithm. So here GA is used to find a good solution initially for TA and TA improves this solution further.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

Allocation and sharing of resources can be done in many ways. But the way to do it using the trust formed by individuals in a social network is being shown here. Here a dynamic cloud-like platform is being created and utilized for sharing of resources. The users are given freedom to decide with whom they wish to share their resources by giving preferences.

. The problem identified in this paper is that this focuses only on one-one allocation system where one requesting user is matched with one providing user. This simple method cannot satisfy complex situations. So the many-many resource allocation should be done. Also the algorithms in the existing system consider only one-one matching. This should be extended for complex scenarios. Further expansion of the model can be done in order for batch allocations and for applying to a complex scenario.

ACKNOWLEDGMENT

We thank God almighty for giving us the strength and blessing us with his grace. We also thank the faculty members and students of our college for their valuable support. We also express our heartfelt thanks to Simon Caton, Christian Haas, Kyle Chard, Kris Bubendorfer and Omer Rana, whose work inspired us to put forward this paper.

REFERENCES

[1] Simon Caton, Christian Haas, Kyle Chard, Kris Bubendorfer, Omer Rana, "A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks," IEEE Transactions on Services Computing, vol. 7. No. 3, July-September 2014.

[2] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social Cloud: Cloud Computing in Social Networks," in 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 99–106.

[3] A. McMahon and V. Milenkovic, "Social volunteer computing," Journal on Systemics, Cybernetics and Informatics (JSCI), vol. Volume 9(4), pp. 34 – 38, 2011.

[4] R. Gracia-Tinedo, M. Sanchez´-Artigas, A. Moreno-Martinez, and P. Garcia-Lopez, "Friendbox: A hybrid f2f personal storage appli-cation," in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012, pp. 131–138.

[5] D. N. Tran, F. Chiang, and J. Li, "Friendstore: Cooperative online backup using trusted nodes," in Proceedings of the 1st International Workshop on Social Network Systems (SocialNet'08), Glasgow, Scot-land, 2008.

[6] C. Haas, S. Caton, and C. Weinhardt, "Engineering incentives in social clouds," in Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011)., 2011, p. 572575.

[7] J. Cappos, I. Beschastnikh, A. Krishnamurthy, and A. T., "Seattle: A platform for educational cloud computing," in The 40th Techni-cal Symposium of the ACM Special Interest Group for Computer Science Education (SIGCSE '09)., Chattanooga, TN USA, 2009.