Priority Based Infrastructure Resources Allocation in Online Social Cloud

Shailesh H. Dinde, Arati M. Dixit

Abstract— **In recent years there has been tremendous growth in online sharing using Social networking and Cloud Computing. In sharing of information there has been pervasive use of Online Social Networking (OSN) websites. Cloud computing has evolved as a paradigm where the computing resources are shifted to the third party to reduce the cost associated with purchase, maintenance and configuration of the resources. In this paper a new paradigm is introduced for sharing of Infrastructure Resources using OSN, which is called Online Social Cloud (OSC). In OSC pre-established trust formed through friend connection between the users in OSN is used. Thus a Social Cloud user can share resources with a friend using pre-established trust on basis of 'give-and-take' policy for mutual benefit. Resources can be easily shared using Social Cloud with security and trust guarantees. Using Social Clouds as a use case in several scenarios supported by co-operative approach exhibits advantages over dedicated infrastructures. Social Clouds appears to be a suitable model for providing core infrastructure as a service on online social computing platforms.**

Index Terms—**Distributed Computing, Cloud Computing, Social Computing, Priority Based resources allocation.**

I. INTRODUCTION

COMPUTING paradigm has experience tremendous growth in last few years. Distributed computing is emerged for sharing of resources for performing high computational task over the network. There are several resources sharing paradigm Volunteer Computing, Grid Computing, P2P computing and Cloud Computing. Cloud computing has evolved as a paradigm where the computing resources are shifted to the third party to reduce the cost associated with purchase, maintenance and configuration of the resources. Cloud Service provider can provide Software, Platforms and Infrastructure as service to user as per requirements of user and payment on basis on usage. In Cloud computing users relieved from the burden of purchasing and maintaining resources; instead resources can be out-sourced to third party service provider and user can get unlimited resources as per requirements. In this paper new the paradigm is Online Social Cloud(OSC). Online Social Cloud is "a resource and service sharing model utilizing pre-established trust between members of a social network" [10]. In this paradigm the merits provided by the conventional cloud computing and extended features of distributed computing paradigms namely the grid computing are combined. Consider the scenario of a computing paradigm where users who collectively construct a pool of resources to perform computational tasks on behalf of their social acquaintance. Social Cloud is similar in many aspects to the conventional distributed-computing. Similarities in this, users can outsource their computational tasks to socially connected peers, and friend will be providing complementarily to their friends for computing using Social Cloud [12]. Most important in connection of OSN is the total computational power provided by users (who are willing to share their idle time and available compute cycles) is free. In this, users are willing to share their computing resources with their friends to form different economical models. This behavior makes this work share commonalities with an existing stream of work on creating computing services through volunteers, although by enabling trust driven from social networks. In Social Cloud the trust is exhibited in social networks as a guarantee for the good behavior of ''workers'' in the system.

***Shailesh H Dinde**, Department of Computer Engineering, PVPIT College, Pune, India.*

***Arati M Dixit** Department of Computer Engineering, PVPIT, Bavdhan. Department of Technology, Savitribai Phule Pune University, Shivajinagar,Pune India.*

II. RELATED WORK

With the increasing nature of online social networking web sites, adoption of social network structures for different types of connection is becoming more common. Key examples are: community and scientific portals like PolarGRID and ASPEN ; social science gateways[1] ; social storage systems like Friendstore , and omemo.com; network and compute infrastructure sharing web sites such as fon.com; models to share insurance policies amongst social peers (friend-surance.de); and where social networks emerge due via collaboration. Ali et al.[2] present the application of Social Cloud model to enable users having less computational power to share access to computation power through platforms like Amazon EC2. In effect they subdivide existing allocations to amortize instance cost over a wider group of users. Using a cloud bartering model (similar to our previous virtual credit model), the system enables resource sharing using social networks without the exchange of money and relying on a notion of trust to avoid free riding. Like writer approach, they use a virtual container (LXC) to provide virtualization within the existing virtual machine instance; however writer approach using Seattle's programming level virtualization provides a much more lightweight model at the expense of flexibility. Mohaisen et al. [3] present an extension to his definition of a Social Cloud computing. Mohaisen et al. investigate how a Social Cloud Computing could be designed, and propose extensions to several well known resource allocation algorithms for task assignments. Their approach considers resource endowment and physical network structure as core factors in the

 All Rights Reserved © 2015 IJARCET

allocation problem, which are different considerations for resource allocation. They analyses the possibility of a Social Cloud via simulation, using several co-authorship and friendship networks as input. In this system a Social Cloud performs based upon variations in load, participation and graph structure.

McMahon and Milenkovic [4] implement Social Volunteer Computing, an extension of conventional Volunteer Computing, where consumers of resources have underlying social relationships with contributor. This approach is similar to the Social Cloud Computing, but it does not consider the actual sharing of resources, as there is no notion of bilateral exchange.

III. MOTIVATION

Volunteer computing is an internet based distributed computing[11], which allows users to share their computational power and helps to run computationally costly projects. Volunteer computing platform consist of millions of users, providing large amount of computational power. Since the rapid growth in the volunteer computing projects, more researchers have been attracted to improve the existing volunteer computing system.

OSN is an alternative approach to establish trust in Cloud computing and Volunteer computing. It is a dynamic environment through which (new) Cloud-like provisioning scenarios can be established based upon the implicit levels of trust that transcend the inter-personal relationships digitally encoded within a social network.

Vision of the Social Cloud[13] is motivated by the need of individuals or groups to access resources they are not in possession of, but that could be made available by socially connected peers which show users are willing to donate personal compute resources to "good" causes. Using this paradigm, users can install a middleware to connect their personal social network, and provide resources to friends, or access resources from their friends. It is considered that resources in a Social Cloud will be shared by user because they are idle, underutilized.

IV. PROBLEM STATEMENT

In Grid computing and Volunteer computing, users donate their idle resources to projects which require high computational power through middleware. But in that framework users are anonymous to each other and are not accountable for their actions. There are two key issues are important here trust and accountability. Goal of Social Cloud is to provide an infrastructure that allows the execution of workflow on traditional Grid resource which can be on demand with additional Cloud Resources, if necessary. This paradigm focused on providing resources to user with trust and proper allocation of resources through priority based resource allocation.

V. THE DESIGN OF OSC

A Social Cloud is "a resource and service sharing framework utilizing relationships established between members of a social network."

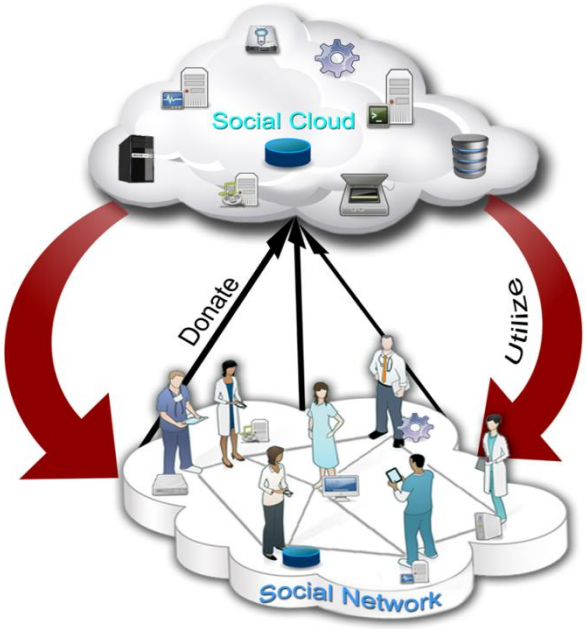

Figure 1. Concept of Social Cloud

A Social network is a set of users and a set of ties representing some relationship among the users. Users in a Social network (people, organizations or other social entities) are connected by a set of relationships, such as friendship, colleague, financial exchanges, trading relations or information exchange. Facebook, Google+ are some examples of Social networking site. Users share their information, thoughts, ideas, photos and videos to friends in online Social network community. Most of the times when users work on their PC, utilize small fraction of computational power hence major section of resources like processor, RAM are in under utilization. Social Cloud aims to provide mechanism for such resource sharing. This model allows users to full computational need of an individual by available underutilized resources of other users in Social network. OSC is a novel concept which considers computational resource sharing standing on the layer of online Social network framework. In Social Cloud, computational resources represent virtualized storage, working environment. Users of Social network take benefit of inbuilt trustful relationship to contribute underutilized resources to others and avails others contributed resources to satisfy personal need. On the foundation of trustful relationship, a user on Social network builds the group. Figure 1 shows Online Social Cloud view standing on Social network, where people donate underutilized computational resources to other users and in return utilizes others resources in need.

A. Architecture of Social Cloud

For implementation of Social Cloud three main areas of functionalities are required defined in Figure 2.

- Outsourcers
- Workers/ Volunteers
- Resource Allocation Server

 All Rights Reserved © 2015 IJARCET

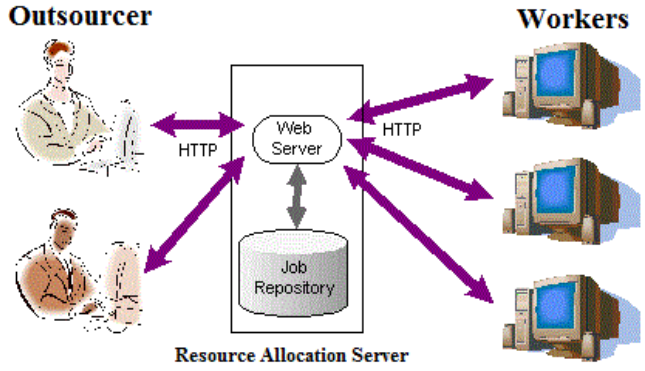

Figure 2. Architecture of Social Cloud

1. Outsourcers

Outsourcers are the users who required high computational power and speed for execute their task on Social Cloud. They divide their task into small chunks and distribute this small works among workers (i.e. Friend on Social Network). Work will distributed by central resource allocation server. Each worker get code for execution and data required for execute that code. After completion of task workers can be submit result to Outsourcer.

2. Worker/Volunteers

Workers are the users who share their computational power (i.e. CPU Cycle, RAM) for outsourcers. Workers share their computational cycle for outsources because they are underutilized or ideal and made available altruistically to "good" cause. For sharing computational power workers need to install middleware at their personal computer. Workers will get task code and data from resources allocation server and compute this task send result back to outsourcers. Scheduling policy for executing task will be depending on operating system of workers.

3. Resource Allocation Server

In design of Social Cloud central resource allocation server is required to collect the local information of workers (i.e. availability of user, current computing power and their workload). Middleware will update this local information periodically to server. In process of resource allocation outsourcers first request for resources then resource allocation server will check available workers and relationship with each other. Worker and outsourcer both can assign priority to each other according to their relationship. The higher priority value gives high preference to user. Assign same priority for different friend is possible. This priority is stored on central resource allocation server. So at the time of resource allocation server will get all this information and provide best match for outsourcer. The allocation server attempts to acquire available resources from the workers to satisfy the request using resource allocation mechanisms. If at the time of reservation the chosen provider is not available then entire process must be re-executed.

VI. RESOURCE ALLOCATION IN OSC

A. Priority Assignment & Resource Allocation

Priority Assignment & Resource Allocation is important requirement for implementation of Social Cloud. In priority assignment, workers and outsourcers both can assign priority to each other. Without it we cannot assume any form of preexistence trust between worker and outsourcer. Allocation in Social Cloud has two different approaches for resource allocation for outsourcer. The decision of resource allocation is whether to centralize or decentralize. The decentralize scheduler impacts the complexity and operations of the entire system. In proposed system centralized resource allocation scheme is used.

- **Resource Allocation Server**

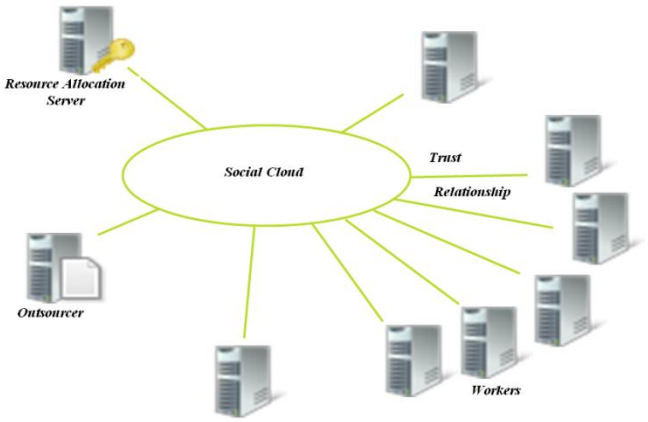

Figure 3. Centralized Resource allocation model

In Centralized Resource allocation (shown in Figure 3) all workers and outsourcers are contact to central server for communicate with each other as well as sharing of resources and outsourcing of task. Each user having middleware for communicate with the server. Middleware can get local information of computing power and update to centralize server after every unit time. So, Central server having all information of currently available user and their computing status.

For example, in order to decide the best set of workers and to outsource task, an outsourcer needs to know which of his friends (i.e. workers) are online and all the information of workers (i.e. availability of resources, online and offline time, computation power) to outsource the task. One possible solution to the problem is to use a centralized server that maintains states. The advantage of centralized resource allocation server is to maintain all local information of the different workers. Instead of communicating directly with each worker, an outsourcer can request for best set of workers among its friends to the centralized resource allocation server. The server can be calculating a set of workers based on the priority assigned to worker and local information of worker.

 All Rights Reserved © 2015 IJARCET

- **Priority Assignment**

In user priority assignment user can specify the priority to their friends according to the relationship among them (friend, colleague, family, etc). This method provides simple preference matching interface in that both outsourcer and worker can define preference or priority for each other. The higher value gives greater priority to their friend. Assigning same value for all friends is possible or grouping is possible. This priority assignment is stored in centralized resource allocation server for allocation purpose. Users also define with whom they are not willing to share or "block user".

In OSC system use a simple numerical priority assignment interface that enables users to define their priority for a friend as both a worker and an outsourcer. The higher value gives greater preference for their friend. A value of 0 or null indicates normal priority and a negative value indicates unwillingness to interact with that friend. When preferences are assigned they are stored in the database and are used to generate the overall priority model for allocation involving the user.

Priority algorithm can be used for generating list of workers for every request of outsourcer. If server receive request for resource then it will be check first available In Priority Based Resource Allocation, when the outsourcer requests for task execution then Resource allocation server gives set of workers for task execution. First server will check higher priority workers of outsourcer if those user's are available then add those user's to list if this user's are not available then it will check other user those having less priority. At the same time workers also check priority for outsourcer. In this resource allocation server will give best workers list to outsourcer depending on priority, availability of user's, workload at worker side. In this case, allocations are computed after certain time intervals, e.g. every half an hour. While this type of allocation computation yields good results for the supply and demand given at the time of the computation, it is unclear what happens in the case of new or changing supply and demand. For example, when users offer/request new resources or retract offers/requests in between two calculation intervals. It is clear that if allocations are only (re)computed at predetermined time intervals, resources will be idle and requests will be left (or become) unsatisfied. As far as all are aware, existing preference based matching literature does not consider such settings. The "optimal" solution is immediate re-matching of the entire supply and demand. In this case, no (new) supply would be idle if an allocation was available, and the resulting allocation would always be stable (assuming we use a stable matching algorithm). However, this places additional requirements on the system. Firstly, computing resources (e.g. VM's) would have to be migratable at any given point in time, and, secondly, the runtime of the allocation mechanism has to be short.

The "worst" solution would be to disregard any new incoming supply or demand until the next time the allocation is computed. In this case, new supply/demand would be idle until the next batch allocation, even if there were corresponding demand/supply. We consider two cases of heuristics for this case:

1) **Random**: allocate resources to/from random friends of the user; and

2) **Greedy**: allocate the best available match for incoming, requester/provider based upon their sharing preferences.

B. Feedback

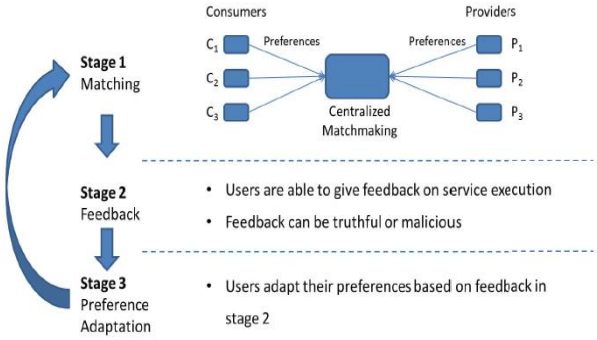

Figure 4. Feedback in priority based resource allocation[14]

Feedback will focus on the point of view of a new user in the system, who wants to use resources from friends or provide resources to friends. Depending on whether user can provide successful or unsuccessful in service. As the user wants to be matched to a consistent user, he can use the trust and decide a preference ranking. The level of trust depends on past relations among the different users in case of direct transactions, and the feedback about previous interactions from other users.

After completion of work user can give the feedback for worker about his experience with him. Feedback will help for new user and existing users for assign or change their preference ranking of friends. If users not directly connected on the social graph, or there were no past transactions between the them, then the preference ranking will depend on the feedback that is provided by other users.

Feedback is an iterative process [14] as in Figure 4. In which after resource allocation workers perform their task and submit result to outsourcer. After completion of work, the users have the chance to give feedback about the previous transaction. In the last stage, users use this feedback to update their preference rankings.

C. Task Generation Policy

Outsourcer will divide his task into different small chunks for distribute his work among different workers (friends). For dividing task into small unit there are two different approaches.

- **Constant Task** : Weight Outsourcer will divide his task into equal size. So, each worker will get equal size of task to compute.
- **Variable Task :** Weight Outsourcer will divide his task into variable in size. So, each worker will get variable size of task to compute.

D. Task Execution Scheme

In Social Cloud paradigm, the goal of the system is to provide resources to outsourcer as per his requirements. In order to fulfill the required computational power requires infrastructure resources from worker. There are two schemes for Task execution.

- **Fixed Contribution**
  For fixed contribution schemes, the worker will decide, how much percentage of his resources (Processor cycle and RAM) will be given for computing the task. Each worker will decide his percentage of contribute for OSN. This will be use

 All Rights Reserved © 2015 IJARCET

later case to calculate the fixed percentages, because it is more considerate of users with low resource endowments, which otherwise would have to allocate most of their resources for the infrastructure.

- **Variable contribution**
  In variable contribution scheme, user will choose their level of contribution based on their individual preferences for resource usage, considering for example altruistic motivation. This scheme addresses the key motivation of an OSC, that is, users voluntarily choose to provide resources to friends.

VII. RESULTS AND ANALYSIS

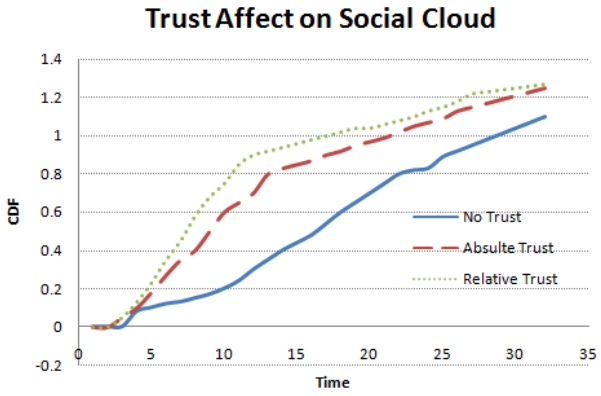

Figure 5. Trust Affect on execution of OSC

The ratio of outsourced computation is depending on the absolute trust by a user towards other user based on preference assignment. It is assumed that the malicious user does not complete task or return any computation results to the outsourcer. Failure of performing computations due to malicious activities when not using trust models. There is less number of malicious activity perform by malicious user in Absolute trust and Relative trust. Using Relative trust user can change there preference and avoid malicious user from performing malicious activity. Fig. 5 shows the outcomes of this experiment. We observe that in both cases where the trust-based resource allocation is used, OSC operates well by outperforming the plain scenario where no trust is used. For example, we notice that while only 45 percent of the compute tasks are finished after 15 unit time when not deploying any trust model, about 80 percent and 90 percent of tasks are finished with the Absolute and Relative Trust models, respectively. The perception behind this performance is simple: whereas the no trust treats neighboring user equally, and tasks of equal size are likely to be outsourced to a malicious neighbor thus degrade the overall time for completing tasks. In both Absolute and relative trust having better performance because of task outsource according to priority.

VIII. CONCLUSION

Social Cloud is a distributed computing service that provides computing resources from social network and allocates resources based on priority, in which user can execute their task on resources provided by their socially connected user. In online social cloud users can discover and trade services contributed by their friends, taking advantage of preexisting trust and relationships between them. Feedback is useful for new user to identify truthful or malicious providers in system.

REFERENCES

[1] W. Wu, H. Zhang, Z. Li, and Y. Mao, “Creating a cloud-based life science gateway,” in E-Science (e-Science), 2011 IEEE 7th International Conference on. IEEE, 2011, pp. 55–61.
[2] Z. Ali, R. Rasool, and P. Bloodsworth, “Social networking for sharing cloud resources,”, Cloud and Green Computing (CGC), 2012 Second International Conference on, 2012, pp. 160–166.
[3] A. Mohaisen, H. Tran, A. Chandra, and Y. Kim, ‘‘Socialcloud: Distributed Computing on Social Networks,’’ Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep., 2011.
[4] A. McMahon and V. Milenkovic, “Social volunteer computing,” Journal on Systemics, Cybernetics and Informatics (JSCI), vol. Volume 9(4), pp. 34 – 38, 2011.
[5] K. Chard, K. Bubendorfer, S. Caton, and O. Rana, ‘‘Social Cloud Computing: A Vision for Socially Motivated Resource Sharing,’’ IEEE Serv. Comput., vol. 5, no. 4, pp. 551-563, June 2011. 6
[6] C. Haas, S. Caton, K. Chard, and C. Weinhardt, “Co-Operative Infrastructures: An Economic Model for Providing Infrastructures for Social Cloud Computing,” in Proceedings of the Forty-Sixth Annual Hawaii International Conference on System Sciences (HICSS). (Grand Wailea, Maui, USA)., 2013.
[7] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, ‘‘Social Cloud: Cloud Computing in Social Networks,’’, Proc. IEEE CLOUD, 2010, pp. 99- 106.
[8] Aziz Mohaisen, Abhishek Chandra, and Yongdae Kim, “Trustworthy Distributed Computing on Social Net-works”, IEEE TRANSACTIONS ON SERVICES COMPUT-ING, VOL. 7, NO. 3, JULY-SEPTEMBER 2014.
[9] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, “Social Cloud: Cloud Computing in Social Networks,” 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 99–106.
[10] K. Chard, K. Bubendorfer, S. Caton, and O. Rana, “Social Cloud Computing: A Vision for Socially Motivated Resource Sharing,” IEEE Transactions on Services Computing, vol. 99, no. PrePrints, p. 1, 2012.
[11] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, ‘‘Seti@home: An Experiment in Public-Resource Computing,’’ Commun. ACM, vol. 45, no. 11, pp. 56-61, Nov. 2002.
[12] Simon Caton, Christian Haas, Kyle Chard, Kris Buben-dorfer , and Omer Rana “A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Net-works” IEEE Transactions On Services Computing, vol 7, no 3, pp. 359 – 372, 2014.
[13] Shailesh H. Dinde, Arati M. Dixit, “On Sharing Infrastructure Resources using Online Social Networks”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 1, pp. 39-43 January 2015.
[14] Christian Haas, Petri, I. ; Rana, O., “Improving Resource Matchmaking Through Feedback Integration”, 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 2014, pp 906 - 914.

 All Rights Reserved © 2015 IJARCET