On Sharing Infrastructure Resources using Online Social Networks

Shailesh H. Dinde[1], Arati M. Dixit[2]

1. Department of Computer Engineering, PVPIT/JSPM, Bavdhan, Pune, Maharashtra, India
2. Department of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India

***Abstract*— With the pervasive nature of Online Social Network (OSN) and Cloud computing, clients are beginning to investigate better approaches to connect with, and use these creating standards. OSNs are digital relationship between users that allows them to share and access information on basis of their social associations. In OSN there is pre-established trust formed through 'friend connections' within an OSN to form a dynamic 'Social Cloud' by empowering users to share resources within a Social Cloud. Users have contributed in making the foundation of social cloud strong via investment in computerized groups. This is further emphasized by the representation, documentation and investigation of social connections. Social Cloud is getting to be more refined. This will facilitate and make it simpler for clients to share their own resources and information via OSN. Thus in a Social Compute Cloud the provisioning of Cloud infrastructure is supported through 'friend' connections is modeled in the proposed model. In particular, the proposed model adventures the trust publicized in OSNs as an issue for the good behavior of other 'workers' in the system.**

***Index Terms*— Social Computing, Distributed Computing, Online Social Network, Cloud Computing.**

1. INTRODUCTION

IN recent years there has been rapid growth in cloud computing and social networking technologies. Cloud computing shifts the computing resources to a third party, eliminating the need to purchase, configure and maintain those resources. Infrastructure as service providers rid users of the burdens associated with purchasing and maintaining computer equipment; instead compute resources can be out-sourced to specialists and consumers can obtain access to an "unlimited" supply of resources. There are two key issues are the notions of trust and accountability between cloud service provider and consumers and providers. In this context, trust and accountability encapsulate several different aspects such as security, privacy, transparency. Goal of proposed system 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. We are focused on provide resource for execution of consumer with proper resource allocation.

A Social Cloud is "a resource and service sharing model utilizing pre-established trust between members of a social network" [10]. In this model, new computing paradigm, called Online Social Cloud (OSC). The proposed paradigm enjoys parts of the merits provided by the conventional cloud and extends features of other distributed computing paradigms-namely the grid computing. Imagine the scenario of a computing paradigm where users who collectively construct a pool of resources perform computational tasks on behalf of their social acquaintance. This paradigm and model are similar in many aspects to the conventional distributed-computing paradigm. It exhibits such similarities in that users can outsource their computational tasks to peers, complementarily to their friends for computing using OSC. Most vital to the connection of Social cloud is the total computational force gave by clients who are willing to share their idle time and available compute cycles. In Online Social Cloud, owners of these computing resources are willing to share their computing resources for their friends, and for a different economical model than in the conventional cloud computing. 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 this paradigm exploits the trust exhibited in social networks as a guarantee for the good behavior of other ''workers'' in the system.

***Shailesh H. Dinde**, P.G. Scholar, PVPIT Bavdhan,Pune,India*

***Arati M. Dixit** Associate Professor, Department of Technology, SavitribaiPhule Pune University, Pune, India .*

2. RELATED WORK

With the increasing pervasiveness of social network platforms, adoption of social network structures for different types of collaboration 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 in developing countries to share access to virtual machines 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

ISSN: 2278 – 1323

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. Mohaisen et al. investigate how a Social Compute Cloud could be designed, and propose extensions to several well known scheduling mechanisms for task assignments. Their approach considers resource endowment and physical network structure as core factors in the allocation problem, which are different considerations for resource allocation. They analyses the potential 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] proposed Social Volunteer Computing, an extension of traditional Volunteer Computing, where consumers of resources have underlying social relationships with providers. This approach is similar to the nature of a Social Compute Cloud, but it does not consider the actual sharing of resources, as there is no notion of bilateral exchange.

3. MOTIVATION

Volunteer computing is a form of internet based distributed computing, which allows users to share their processing cycles, and helps to run computationally costly projects. In existing volunteer computing platforms consist of millions of users, providing large amount of processing cycles and memory. Since the rapid growth in the volunteer computing projects, more researchers have been attracted to study and improve the existing volunteer computing system.

In this paper we argue an alternative approach to establish trust and accountability in Volunteer computing and Cloud platforms: a Online Social Cloud. 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 OSC 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 connected peers which show users are willing to donate personal compute resources to "good" causes. Using this approach, users can download and install a middleware connect their personal social network, and provide resources to, or consume resources from, their friends through a Online Social Network (OSN). We anticipate that resources in a Social Cloud will be shared because they are underutilized, idle, or made available altruistically.

4. SOCIAL COMPUTING DESIGN AND ARCHITECTURE

An Online Social Cloud is "a resource and service sharing model utilizing pre-established trust between members of a social network."[10].

The expanding depth of social networks has prompted a world in which numerous relationships and cooperation's are also represented on the web. These social digital relationships have created new opportunities to define socially oriented computing paradigms such example is the

1) Social Cloud computing model.

2) A collaborative resource allocation model built upon a social network.

An Online Social Compute Cloud is intended to empower access to flexible figure abilities gave through a cloud fabric built over resources provided by socially connected users. An OSC is provided virtualized resources that expose (secure) access to contributed resources, i.e. CPU time, memory and disk/storage of user through this they are able to execute programs.
Vision of the Online Social Cloud 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 connected peers. OSC present a infrastructure resource allocation using social connection of user. Using this approach, users can download and install a middleware, leverage their personal social network via a Social application, and provide resources to, or consume resources from, their friends. We anticipate that resources in a OSC will be shared because they are underutilized, idle, or made available altruistically. However a key aspect of a Social Cloud is the notion of sharing, not selling, resources. Specifically, due to the social network basis of a Social Cloud, users will have explicit preferences with whom their resources are allocated to, and from whom they consume resources.

4.1 Design **of Online Social Cloud**

The main design of Online Social Cloud is very simple. OSC provide bulk of inexpensive resources that supply more computing power to user. This will offer scalable, reliable and powerful computing platform to users where their task are divided into several small unit and distributed among different workers (i.e. friends of outsourcer). User can outsource their computational task to his friend. In order to perform this different task at outsourcers and workers side middleware is needed which collect chunks (a portion of code and data to compute) from outsourcers and send this chunks to workers for processing and effective resource allocation for outsourcing tasks. Once task is outsourced to given worker (both code and data), the worker is left to decide how to schedule and execute the task locally to compute it. Worker will perform the required computation based on received data and code and send computational result to outsourcer.

ISSN: 2278 – 1323

4.1.1**Scheduling Entity**

In the Social Cloud, two different types of schedulers can be used for resource allocation [8]. One of this is used at worker side for how tasks computed and in which order, and another scheduler is for task outsourced to each worker (i.e. resource allocation). The decision used for outsourced whether to centralize or de-centralize the former scheduler impacts the complexity and operation of the entire system. In the following, elaborate on both design decisions.

- **Decentralized scheduler**

  This scheduler responsibility of outsourcer is to decide scheduling of out-sourcing tasks. Outsourcer will get information from all workers those are currently available for computing in a decentralized manner, thus each node takes care of scheduling its tasks. This could reduce the burden of the design scheduling server in a centralized alternative. But, this could increase the complexity of outsourcer because outsourcer needs to check all information of workers (i.e. availability of resources, online and offline time, computation power). After getting this information outsourcer will decide for outsourcing task among online workers. These information is important for outsourcer while outsource the task to worker for improve reliability, efficiency, result. An illustration of this design option is shown in Figure 1. In this scenario, each outsourcer, as well as worker, has its own separate scheduling component.

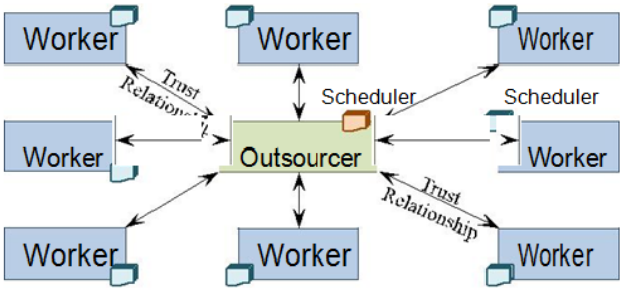

Figure 1. The Decentralized model of task scheduling [8].

- **Centralized Scheduler**

  In centralized scheduler there is centralized server which having all the local information regarding the workers those are online. So the use of a centralized scheduler might be necessitated to reduce outsourcer workload.

  For example, in order to decide upon the best set of workers to which to outsource task, a outsourcer needs to know which of its friends (i.e. workers) are online, all in-formation of workers (i.e. availability of resources, online and offline time, computation power). For that purpose, and given that the underlying communication network topology may not necessarily have the same proximity of the social network topology, the protocol among nodes needs to incur back and forth communication cost.

  The advantage of centralized server is that to maintain local information of the different workers. Instead of communicating directly with workers nodes, an outsources would request the best set of workers among its friends to the centralized scheduling server. The server will produce a set of workers, based on the local information. Such candidates would typically be those that would have the most available resources to handle the outsourced computation task.

  An illustration of this design option is shown in Figure 2. In this design, each node in OSN would periodically send local information to a centralized server. When outsourcer required resources then contacts the centralized server for getting best set of candidates for outsourcing task.

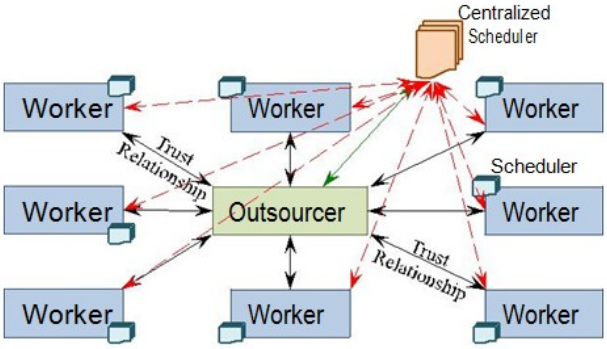

Figure 2. The Centralized model of task scheduling [8].

4.1.2**Tasks Generation and Weights**

There are different approaches for the tasks divided by each outsourcer. The size of each generated task is measured by units of time. There are two different scenarios:

- **Constant task weight**: Outsourcer divides his task into a same size. So, each task having same size and outsourced among several workers in the OSN. The size of each task is T.
- **Variable task weight**: Outsourcer divides his task into a variable task size. The size of tasks as a uniformly distributed random variable in the size.

4.1.3**Contribution Schemes**

Considering the OSC paradigm, it can be argued that the goal of such a system should be to provide sufficient resources to outsourcer to execute his task. In order to acquire the required infrastructure resources by the workers, OSC consider the following two contribution schemes [6]:

- **Fixed contribution**

  For fixed contribution schemes, the worker will decide for how much percentage his resource (Processor cycle and RAM) for computing the task. Each worker will decide his percentage of contribute for OSN. This use the latter 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.

ISSN: 2278 – 1323

- **Variable contribution**

  In variable contribution scheme, users 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.

4.1.4 User Preferences & Resource allocation

This is an important requirement for an Online Social Cloud, as without it we cannot assume any form of preexistence trust between outsourcer and worker. Once the social network of a user has been accessed and the social database populated, the question is how to interpret the user's social ties for the purposes of allocation. There is no single unified methodology for the interpretation of social ties, and which to use is often context dependent.

- **User Preferences**

  In user preferences user can specify the ranks to their friends according to their relationship among them (friend, family, etc). We provide simple preference matching interface in that both outsourcer and worker can define preference for each other. The higher value gives greater preference to their friend. Assigning same value for different friends is possible. This preference assignment is stored in centralized server for resource allocation. Users also define who they are willing to share with, or "block" users.

- **Resource Allocation**

  Resource Allocations based on the principle of best effort and random allocation. When allocating resources the Resource Allocation Server filters the list of donated resources. The general process of allocation in the Resource Allocation Server is to first determine available donations with whom the requesting user has a relationship. To do this the list of all donations in the system is filtered by the list of friends for a particular user. The outsourcer's preferences for each possible friend are then computed by retrieving preferences stored in the database. Likewise the preferences for each of these friends for the requesting user as an outsourcer are computed. This information is then aggregated and sent to the matching service to determine an appropriate match. The Resource Allocation Server attempts to acquire available nodes from the provider to satisfy the request using resource acquisition mechanisms. If, by the time of reservation, the chosen provider is no longer available the entire process must be re-executed.

5. PROPOSED SYSTEM ARCHITECTURE

Building upon Online Social cloud we leverage the same base implementation for account creation and registration processes, donation infrastructure, and resource allocation mechanisms. Proposed system extended and deployed a new Resource Allocation Server that leverages social information derived from users and their relationships.

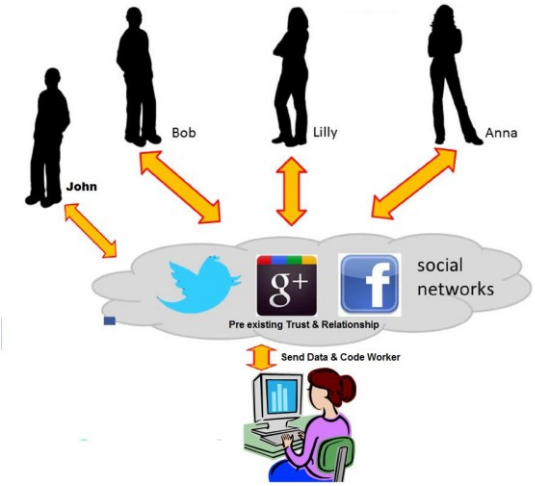

Figure 3. A Social Compute Cloud .

Figure 4 define three areas of functionality needed for the construction of an Online Social Cloud:

1) **Resource Allocation Server**: The server for store the local information of user as well as their relationship. Outsourcer can contact to Resource allocating server for best set of worker for executing task.
2) **Volunteers/Workers**: User those who are contribute their Computing cycle for outsourcer.
3) **Clients/Outsourcer**: User those who are required high computing speed for execute their work.

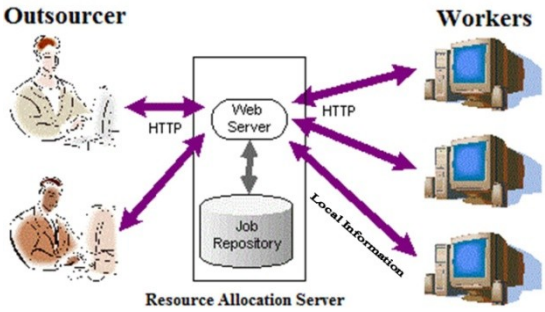

Figure 4. Proposed Architecture of OSC

5.1 OSC Resource Allocation Scheme

5.1.1 Online Social Network

In order to access users' profile information and relationships, the Resource Allocation Server requires access to a user's Social profile. To do so, we have created a Social application for the social clearing house that requests access to profile information, friends and friend lists of registered users. The Social application is integrated with the Resource Allocation Server. Authentication with

ISSN: 2278 – 1323

Social application is uses their username and password. The Resource Allocation Server stores this access token when a user logs into the service and uses it with the Social APIs to obtain the profile and friend lists. The Resource Allocation Server stores the list of friends for each user in an application database and periodically refreshes this information.

5.1.2Preference Assignment

In this system use a simple numerical preference matching interface that enables users to define their preference for a friend as both an outsourcer and a worker. The higher value gives greater preference for their friend. A value of 0 or null indicates no preference and a negative value indicates unwillingness to interact with that friend. When preferences are assigned they are stored in the application database and are used to generate the overall preference model for allocation involving the user.

5.1.3Social Resource Allocation

The general process of allocation in the Resource Allocation Server is to first determine available donations with whom the requesting user has a relationship. To do this the list of all workers in the system is filtered by the list of friends for a particular user. The consumer's preferences for each possible friend are then computed by retrieving preferences stored in the database. Like-wise the preferences for each of these friends for the requesting user as a consumer are computed. This information is then aggregated and sent to the matching service to determine an appropriate match. The Resource Allocation Server attempts to acquire available nodes from the provider to satisfy the request using resource acquisition mechanisms.

6. CONCLUSIONS

In attempted methodology the architecture and design of an Online Social Network; a combination of Cloud Computing, Volunteer Computing and Social networking. In online social cloud users can discover and trade services contributed by their friends, taking advantage of preexisting trust and relationships between them. Using this approach, users are able to execute programs on virtualized resources provided by their friends. This methodology used by users to communicate with each other and interact with the resources of their friends.

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] 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.

ISSN: 2278 – 1323