A Social Compute Cloud: For Sharing Resources

Surjimol R[1]

[1] M.Tech Student, Dept. of Computer Science and Engineering,
M.G University, Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala, India

Abstract: ***Social networks platforms changed the way of people's interaction. They promoted the establishment of digital communities as well as the annotation, documentation and exploration of social relationships. Since application softwares are more complicated, it enable users to share their own services, resources and information via social networks. A Social Compute Cloud allows the provisioning of Cloud infrastructure occurs through "friend" relationships. In a Social Compute Cloud, resource providers possess virtualized containers on their personal computers or smart devices to their social network. Often, Users may have complex preference structures regarding with whom they want to share their resources, By proper simulation, resources can be effectively allocated within a social community offering resources on a qualitative basis. The main advantage is that social networks can be used in the implementation of cloud computing infrastructures and the resources can be allocated in the presence of user sharing preferences.***

Keywords: Preference structure, social compute cloud, resource allocation, social network**.**

1. Introduction

A Social Cloud can be considered as a cloud platform where resource and service sharing framework utilizing relationships established between members of an existing digital community is followed. The social cloud offers Cloud-like provisioning of services which is based on the notion of trust and accountability between resource consumers and providers which encapsulate several aspects such as protection of rights, security of services and other issues. Provider certification and service level agreements address the main issues of trust and accountability.

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 Social Cloud. 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 Social Cloud, owners of these computing resources are willing to share their computing resources for their friends circle, 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. This paradigm exploits the trust exhibited in social networks as a guarantee for the good behavior of other workers in the system.

The main feature of a Social Cloud is that it enables sharing, not selling of resources. Due to the social network dependence of the Social Cloud, users have explicit structural preferences with whom their resources are allocated to and from whom they consume resources. To support user preferences, Bidirectional preference-based resource allocation algorithm is used.

2. Related Works

There are multiple instances of Social network and Cloud computing integration. However, most examples use Cloud platforms to host Social networks or create scalable applications within the Social network. Key examples are: community and scientific portals like PolarGRID and ASPEN (Automated Service Provisioning Environment); ASPEN takes an enterprise approach to integrating Web 2.0, Social networking and Cloud Computing by exposing applications hosted by Cloud providers to user communities in Facebook. PolarGrid is one such example which extracts Social data using the OpenSocial interface and relies on OpenID for identification.

McMahon and Milenkovic proposed Social Volunteer Computing, an extension of traditional Volunteer Computing, where consumers of resources have underlying social relationships with providers. Bilateral exchange is not possible in this type.

Ali et al. proposed the application of a Social Cloud model to enable users in developing countries to share access to virtual machines through platforms like Amazon EC2. Existing allocations is subdivided to reduce instance cost over a wider group of users. Using a cloud bartering 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. It uses a virtual container to provide virtualization within the existing virtual machine instance.

Mohaisen et al. proposed an extension to the definition of a Social Cloud. This approach considers resource endowment and physical network structure as core factors in the allocation problem, which are the different aspects for resource allocation. The potential of a Social Cloud is analysed by simulating several co-authorship and friendship networks as input. It analyzes how a Social Cloud performs based upon variations in load, participation and graph structure.

Tan et al. motivated the philosophy of a Social Cloud with the core ideas of sharing and exchanging resources within a social network or community to tackle Big Data problems.

Gracia-Tinedo et al .proposed a Friend-to-Friend Cloud storage solution like F2Box. It retains a reliable service while using the best effort provisioning of storage resources from friends. Since a pure friend-to-friend system cannot compare in terms of quality of service with traditional storage services. So a hybrid approach where reliability and availability can be improved using services like Amazon's S3 provides a valuable consideration in the realisation of a Social Cloud.

Kuada and Olesen propose opportunistic cloud computing services (OCCS): a social network approach for the provisioning and management of enterprise cloud resources. It provide a governing platform for enterprise level social networking platforms consisting of interoperable Cloud management tools for the platform's resources, which are provided by the enterprises themselves.

Gayathri et al. and Chen and Roscoe proposed the security constraints in the construction of a Social Cloud. They provide counter measures for, how a Social Cloud can be used to prevent copyright and other unfavorable actions that violate security.

3. Motivation

Volunteer computing is a form of internet based distributed computing, which allows users to share their processing cycles, and helps to run mathematically high cost projects. The existing volunteer computing platforms provide large amount of processing cycles and memory for millions of users.

In this paper we argue an alternative approach to establish trust and accountability in Cloud platforms: a 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.

The view about 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 which show users are willing to donate personal compute resources to "good" causes.

Using this method, users can download and install a middleware connect their personal social network, and provide or consume resources to or from their own friends through a Social clearing house. We expect that resources in a Social Cloud will be shared because they are idle, unutilized or made available randomly.

4. Design of Social compute cloud and its architecture

A Social Cloud is "a resource and service sharing model utilizing pre-established trust between members of a social network".

The expanding depth of social networks has prompted a world in which numerous relationships and their associations are also represented on the web. These social digital relationships have created new opportunities to define socially oriented computing paradigms such example is the Social Cloud computing model. It is a collaborative resource allocation model built upon a social network.

A 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. A Social Cloud 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 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. Social compute cloud 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. The key aspect of a Social Cloud is the concept of sharing, not selling, resources.

4.1 Design of Social Compute Cloud

The main design of Social Cloud is very simple. Social Cloud 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 i.e. a portion of code and data to compute from outsourcers and send this chunk 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.

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. The decision used for outsourced whether to centralize or de-centralize the former scheduler impacts the complexity and operation of the entire system.

In decentralized 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.

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. 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.

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:

- By Constant task weight: Outsourcer divides his task into a same size. So, each task having same size and outsourced among several workers in the Social Network. The size of each task is T.
- By 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 Social Cloud 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 social network. 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.
- 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.

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 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 which 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

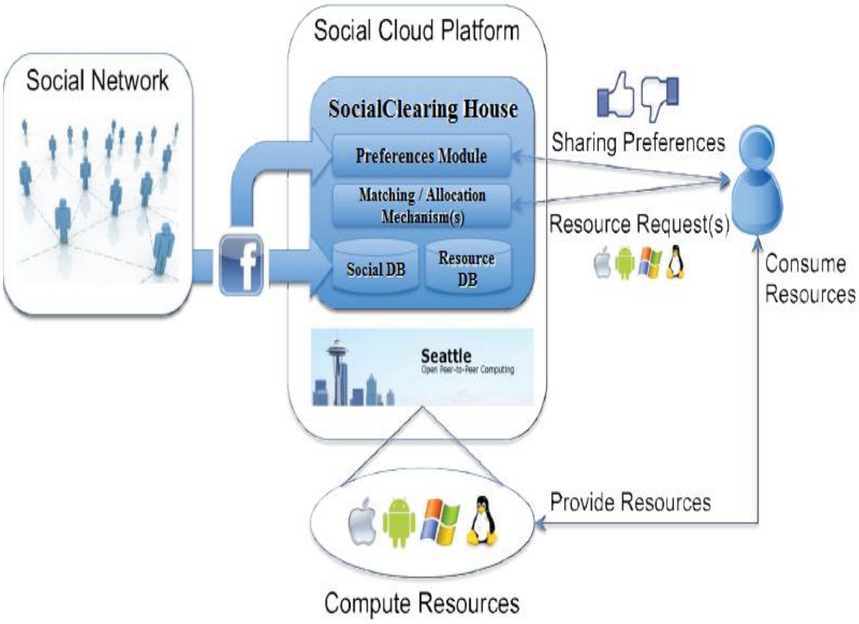

Figure 1: Social Compute Cloud and its Core Components

Building upon Social cloud we use 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.

The functionalities needed for the construction of a Social Compute Cloud:

1) A Social Cloud Platform: It is the technical implementation for the construction and facilitation of the Social Cloud as well as necessary middleware to enable resource sharing between "friends" at the edges of the Internet.
2) A socio-technical adapter: The means to observe and interpret social ties for the elicitation or derivation of sharing preferences.
3) A socioeconomic model: the formulation of a social microeconomic system for the allocation of resources upon the premises of social ties, and preferences with respect to how social ties denote a user-specific willingness to consume and/or provide resources.

5.1 A Social Compute Cloud Platform

The cloud platform coordinates and facilitates its basic functionalities such as user management, resource allocation, etc. A Social Clearing House defines how supply is allocated to demand. a social clearing house captures the following: the protocols used for distributed resource allocation, the rules of exchange land the formalization of one or more allocation mechanisms. A social clearing house is therefore the central point in the system where all information regarding users, their sharing preferences to access the friend circle, and the details about resource supply and demand is stored. Due to this the social clearing house requires two databases: One to capture the social graph of the current users and their sharing preferences. The second one act as a resource manager to keep track of resource acquisition, availability, and allocations.

A middleware to provide the basic resource availability, virtualization of resources and sandboxing mechanisms for provisioning and consuming resources. It defines a set of rules needed for users and resources to interact with the system. For these purposes a middleware called Seattle is selected. Seattle is extended as it cannot allocate its resources based upon digital relationship.

A socio-technical adapter, which can be considered as an API which is used to integrate users' social network to a cloud platform.Moreover it acts as a means of authentication. To facilitate resource allocation the social clearing house requires the sharing preferences of the user once their social network has been accessed via the socio-technical adapter. Therefore, a preferences module is provided for capturing and representing the sharing preferences.
Matching Mechanisms are socio-economic implementations of the social clearing house microeconomic system. They determine appropriate allocations of resources via users' sharing preferences across their social network. Compute Resources are the technical details of users which they provide to or consume from the Social Cloud.

5.2 Implementing Social Clearing House

5.2.1 Social Network Integration

In order to access user's profile information and relationships, the Resource Allocation Server requires access to a user's Social profile. The Social application for the social clearing house that requests access to profile information of registered users. The Social application is integrated with the Resource Allocation Server. Authentication with Social application uses their username and password. The Resource Allocation Server stores the 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.

5.2.2 Preference 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.

5.2.3 Social Resource Allocation

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

6. Conclusion and Future Works

In attempted methodology the architecture and design of a Social compute cloud; a combination of Cloud Computing, Volunteer Computing and Social network. In a 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.

As future work, additional ways can be included for users to provide their sharing preferences and methods which automatically identify them from their social network by using clustering based on relationship lists and relationship strength. In Social Cloud platform we can extend the sandbox to provide additional functional system calls and social permission control so that users can give access permission rights to groups. This will increase the number of applications that could be executed within the Social Cloud and also extends the social integration of the system. Thereby, moving our implementation to a productive 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, “Social cloud: 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.

Author Profile

Surji mol R received the B.Tech degree in Computer Science and Engineering from Kerala University, Kerala at Sree Buddha College of Engineering, Pattoor in 2012. And now she is doing her M.Tech degree under M.G university, Kerala in Mount Zion College of Engineering.