

Incentivising Resource Sharing in Social Clouds

Magdalena Puncea*, Ivan Rodero*, Manish Parashar*, Omer F. Rana[†] and Ioan Petri[‡]

*Center for Autonomic Computing, Rutgers University, NJ, USA

{puncea, irodero, parashar}@cac.rutgers.edu

[†]School of Computer Science & Informatics, Cardiff University, UK

o.f.rana@cs.cardiff.ac.uk

[‡]School of Engineering, Cardiff University, UK

PetriI@cardiff.ac.uk

Abstract—Social Clouds provide the capability to share resources among participants within a social network – leveraging on the trust relationships already existing between such participants. In such a system, users are able to trade resources between each other, rather than make use of capability offered at a (centralized) data centre. Incentives for sharing remain an important hurdle to make more effective use of such an environment, which has a significant potential for improving resource utilization and making available additional capacity that remains dormant. We utilize the socio-economic model proposed by Silvio Gesell to demonstrate how a “virtual currency” could be used to incentivise sharing of resources within a “community”. We subsequently demonstrate the benefit provided to participants within such a community using a variety of economic (such as overall credits gained) and technical (number of successfully completed transactions) metrics, through simulation.

I. INTRODUCTION & CONTEXT

Over recent years there has been a considerable increase in the availability of mobile devices with internet capability, along with a proliferation of set-top boxes and home user machines (such as media centres) supporting content download. This can also be observed by the significant increase in the take up of TV-over-IP (TVoIP) services – according to a survey from Samsung electronics ¹, Internet video (which accounts for television, Video-on-Demand and Peer-2-Peer traffic) will account for 91% of consumer Internet traffic in 2013. Similarly, according to other surveys (such as from Morgan Stanley ² conducted in 2010), mobile will be bigger than desktop internet in 5 years. Such studies also demonstrate the considerable new modes of service delivery and access that will become available when mobile internet becomes the dominant mode of access to content. It is also inevitable that such mobile devices (including set-top boxes) are not just used to access content, but also provide caching, storage and processing capability to other users (limited currently by battery power and security considerations).

In previous work, the STACEE system [1] was developed for dynamically creating storage Clouds using edge devices, such as routers, routing switches, multiplexers, mobile phones, PCs/media centers, set-top boxes and modems. The functional

architecture within STACEE makes use of edge device capacity in a Cloud using Peer-to-Peer (P2P) technology, thereby reducing energy consumption at a single site and maximising user engagement with the system. With a four layer architecture, the system ensures a high level of communication, control, synchronization and data access. The availability of storage resources and access latency are also significantly improved. The establishment of such P2P-based community Clouds requires a robust mechanism for controlling interactions between end-users and their access to resources/devices. For instance, in the context of such a Cloud model, end-users can contribute with their own resources in addition to making use of resources provided by others (at different times and for access to differing services). It is necessary to have inherent trust between users and providers of such systems, an approach also explored in Social Clouds [2] – which considers how storage resources can be traded between *friends* within a social network such as Facebook. The Social Cloud model assumes the existence of trust between users within such a system, thereby overcoming some of the restrictions to sharing that would arise within a more open environment. A number of incentive models are also outlined in [2] that encourage users to contribute resources for use by others. In previous work [3], we have also investigated the role of trust relationships between participants within a Social Cloud, in particular identifying how malicious behaviour (through inaccurate feedback) could impact the selection of a suitable resource provider.

In this paper, we describe how a resource sharing model based on modelling intra- and inter-community exchange can be supported within a P2P, edge-device based environment. We propose a decentralized model that provides: (i) incentives for providers to offer resources; (ii) consumers to provide feedback about the quality they receive. We argue that this model provides a more natural model for sharing resources – as it does not rely on any central coordination servers or the establishment of a central exchange currency. Our model is based on the concept that excess capacity owned by a user can be traded within a “community”, in order to earn virtual currency and credits they could utilize at a later time. Our model is validated through simulation and two design implementations based on the CometCloud framework are proposed.

¹<http://mpeg.chiariglione.org/tutorials/seminars/mmt-2010/Samsung.pdf>

²http://www.morganstanley.com/institutional/techresearch/pdfs/Internet_Trends_041210.pdf

II. SHARING MODELS

Our approach is based on the concept of *local* communities identified by the socio-economist Silvio Gesell in his book “The Natural Economic Order” [4]. In local communities participants can generate their own currencies and exchange services between each other, thereby contributing to a more autonomous market exchange. The approach utilizes demand within a community to set up local service providers. “The Natural Economic Order” identifies sharing models between participants within a regional context, building on previously established trust relationships, and providing a process for the exchange of goods and services with the ultimate objective of regulating a specific market. Gesell’s approach also includes other aspects (not considered here), such as a currency losing value over time and therefore the need to support currency circulation and prevent hoarding.

In our model, every node (or exchanging entity) has its own currency – the value of which fluctuates according to the reputation of the node as a provider (based on its ability to offer a capability) within a market. We assume participants in the network belong to a community (modelled as a “cluster”) such that nodes that belong to the same community/cluster are connected directly or by a few hops in the underlying social graph. The notion of a cluster is based on the observation that often nodes that are connected through social relationships are more likely to interact and trust each other. Therefore clusters are characterized by a certain level of social trust. When one node is using a service from another, a *transaction* occurs between them. More formally a transaction is defined by a consumer, a provider and a price.

In our approach, we define *two payment schemes* between pairs of nodes. Payment within the same cluster involves converting a currency of one node with another – depending on the current exchange rate known within the cluster. Currency exchange rates depend directly on a node’s reputation. Nodes that belong to different clusters are not allowed to convert currencies i.e., payments are kept only as credits, similar to credit networks [5], [6], [7], [8]. Payments in credit networks involve passing IOU (I Owe You) messages between trusting nodes. When a node A offers a service to node B, then B passes an IOU to A, indicating that node A can receive a service in the future from B (and only from B) in return.

Consider that during a transaction node A (using A-dollars) pays node B (which uses B-dollars) – then there are two possibilities:

- 1) If A and B are in the same cluster, B will transform A-dollars into B-dollars according to the current exchange rate. If $rep(A)$ is the reputation of node A and $rep(B)$ the reputation of node B then 1 A-dollar equals $rep(A)/rep(B)$ B-dollars. B can then use its B-dollars to buy services from other nodes in the network and is not restricted to trade with A only (i.e., if it kept A-dollars as credit) – unlike credit networks. It is therefore necessary to maintain exchange rates to enable such conversion to be carried out.

- 2) If A and B are in different clusters: B will keep A-dollars and use these as a credit with A for a future transaction.

When two nodes belonging to the same cluster successfully complete a transaction, the consumer node gives feedback (as a score) about the provider node – leading to an update of the reputation of the provider node. A reputation value is calculated as an average of all received scores. In every cluster, a list of *reputation values* for nodes that belong to that cluster is maintained.

Figure 1 provides an illustrative example of two clusters. It shows the role and currency of each node involved in two transactions and the flow of messages between nodes in *Transaction 1*. When a service is requested and delivered, the consumer pays for it (x A-dollars) and scores the provider’s service. Specifically, in step (5) the provider’s reputation is updated by combining the new score with previous values. This could be undertaken in a number of possible ways: (i) averaging of values across all previous transactions – which is the simplest approach that could be adopted; (ii) weighting of values based on recency – i.e., transactions that have completed most recently are given a greater weighting than those that occurred a long time ago – this ensures that good recent behaviour is given greater emphasis; (iii) normalising the weighting across all nodes within a system, similar to Eigentrust [9]. Normalisation is particularly useful to avoid giving very high reputation scores to particular nodes and thereby always selecting them, limiting selection of other nodes (in this way, although a normalisation approach is used similar to Eigentrust, it is different objectives). Reputation values can be stored by multiple cluster header nodes or alternatively a distributed hash table (DHT) can be used to store and retrieve them.

Transaction 2 reflects the service payment in consumer’s currency (A-dollars), which cannot be converted to the provider’s currency (B-dollars). Such a transaction demonstrates cross-cluster interaction, a useful outcome to ensure that nodes within clusters are still able to collaborate, but with credits that can only be redeemed by one provider. The ability to support both currency conversions and credits increases the incentives to collaborate and enable both “trustful” sharing (within a cluster) and “explorative” sharing (between clusters).

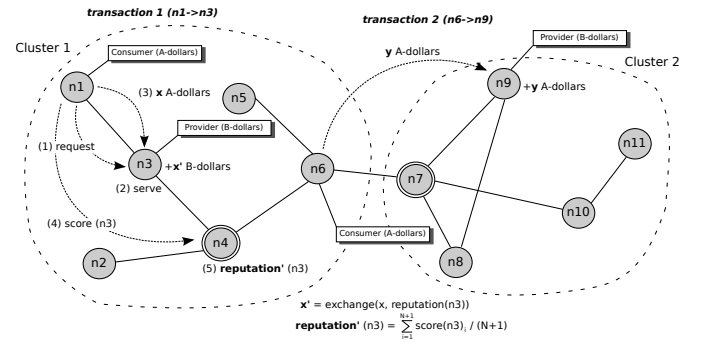


Fig. 1: Example transactions. Transaction 1 is performed within a cluster and transaction 2 between different clusters

By “trustful” sharing we assume that nodes in a cluster report scores about providers/resources honestly and convert currencies according to the cluster related exchange rates. As a part of our future work we plan to consider and analyze non-cooperative behavior as well.

III. EVALUATION

To demonstrate the benefits and limitations of using the resource sharing model we perform a set of experiments taking account of the connectivity within and between clusters. Our experiments answer the following questions: (i) how does the number and sizes of clusters/communities affect the number of transactions completed? (ii) is the approach scalable? (iii) how much does the non-uniform (power-law) distribution of the reputation and social graph degrees affect successfully completed transactions?

A. Experimental Setup

All nodes are connected via an underlying social graph. We simulated a social graph using the findings from a large study about the Microsoft Messenger instant messaging (IM) system [10]. The study identifies that the graph has two important structural properties which are related to the: *node degree distribution* and the *clustering coefficient*. In this study, two graphs from the instant-messaging system were derived and observed: a buddy (friends) graph and a communication graph. The buddy graph contains edges between nodes who are buddies or friends i.e., are in each others contact lists while the communication graph contains edges only between those buddy nodes who have communicated during the observed period. Both graphs have a power law node degree distribution with an exponential cutoff $p(k) \approx k^{-a}e^{-bk}$. It was observed that the communication graph has parameters $a = 0.8$ and $b = 0.03$ and the buddy graph has $a = 0.6$ and $b = 0.01$. The second property is related to the clustering coefficient, which is a quantitative measure of transitivity in a communication network i.e., showing to what extent people who have common friends tend to be friends themselves. The clustering coefficient of a node is formally defined as a fraction of triangles around a node, where a triangle refers to the situation when two friends of a node are friends themselves. Some earlier studies on web graphs [11] show that clustering coefficient decays as $c = k^{-1}$ while the study with the instant-messaging system shows it decays at a much slower rate, as $c = k^{-0.37}$. This indicates that the clustering coefficient in the social graph is higher than in the web graph. The average clustering coefficient was observed to be 0.137. We used these two structural properties to simulate an online social network graph. Our simulated graph has a node degree distribution $p(k) \approx k^{-a}e^{-bk}$ with $a = 0.8$ and $b = 0.03$ which corresponds to social communication graph. We tuned the average clustering coefficient to be ≈ 0.137 .

Subsequently, we simulated a clustering algorithm that divides the nodes of the graph into a large number of small clusters. Our clustering algorithm satisfies two conditions: (1) clusters are non-overlapping and (2) nodes that belong

to the same cluster are all connected. We used merging of neighboring clusters to experiment with varying numbers of clusters and different cluster sizes. We carried out various experiments with a pre-defined node consumer, provider and a price. A transaction is carried out *along the shortest path's chain of nodes* between the consumer and the provider. After a transaction the node consumer gives a score about the provider node based on: (i) randomly from a predefined interval for scores; (ii) a distribution (following a non-uniform power law) based on the Quality of Service, e.g. if a quality value for a node provider is q , then the score will be randomly selected over the interval $[q - t, q + t]$ where t is a threshold.

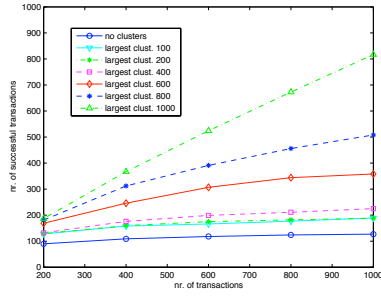
B. Experimental Results

We performed a set of experiments to study the long term liquidity (i.e., capacity to route payments when nodes repeatedly transact with each other), as well as the overall credits gained during transactions. We analyzed how various parameters influence the success of transactions. In particular, we observed the following parameters: (1) the size and the number of clusters, (2) the distribution of reputation values, (3) the size of the social graph, (4) the density of the social graph and (5) the limit of nodes' accounts.

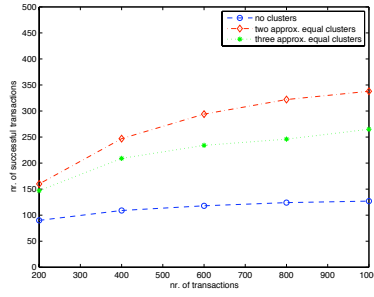
1) *Impact of cluster sizes*: Assuming that “socially close” nodes trust each other, our first experiment aims to identify how much it pays to trust each other. For this purpose we observed the network under the following conditions: when no trust exists at all, when trust exists within small groups and large groups. The main metric used is the number of successful transactions. A transaction is successful if every single payment between a pair of nodes along the chain between the provider and the consumer is successful. Our purpose is to find out how much the success of transactions increases when we increase the clusters' sizes.

The number of nodes in our initial setting is 1024 and the maximal degree a node could have is 64. Note however that node degrees are selected according to the power-law distribution and most of the nodes have degrees much less than 64. In each experiment we start with a predefined set of transactions to be attempted. We present results for five experiments with 200, 400, 600, 800 and 1000 transactions respectively. For each transaction we specify a node consumer and a node provider which are selected uniformly at random from the set of all nodes. The value of the price was set to 20 and the account limit for each node was set to 100. These parameters were selected to result in some nodes being depleted of their budget/account. This happens when the number of transactions a node can sustain defined by the ratio of account and price is less or equal to the number of transactions a node participates in. In this experiment all reputation scores were selected uniformly at random from the interval from 0 to 10.

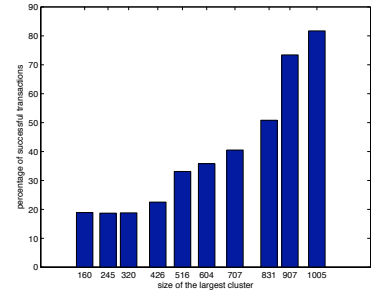
Figure 2a illustrates the number of successful transactions for the different clustering scenarios. In each of these different scenarios we have one large cluster and several smaller clusters. We illustrate how the number of successful transactions



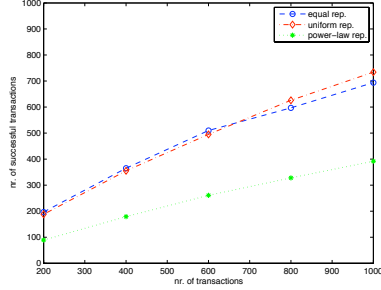
(a) Impact of clusters



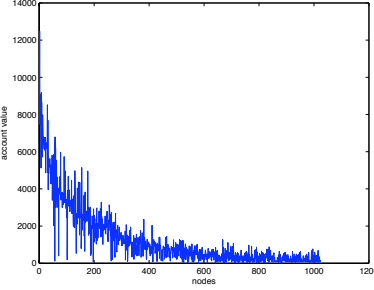
(b) Impact of clusters: two and three large clusters



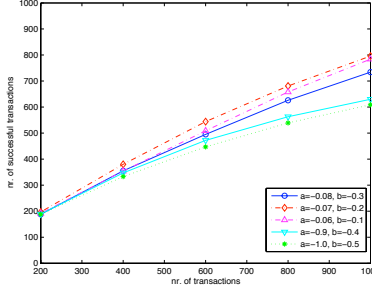
(c) Percentage of successful transactions depending on largest cluster



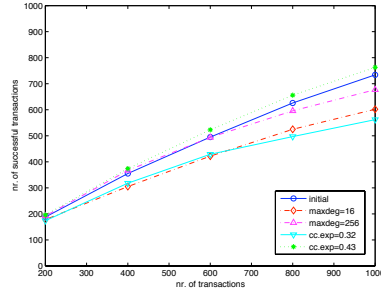
(d) Reputation distributions



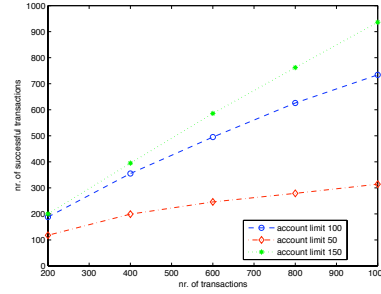
(e) Account values



(f) Impact of the power-law distribution



(g) Impact of the clustering coefficient exponent and maximal node degree



(h) Impact of the limit of accounts

Fig. 2: Summary of experimental results

differ depending on the size of the largest cluster. We compare these different scenarios to the case with no clustering which corresponds to the credit networks. First observation is that although the success rate increases with increasing the largest cluster size, it increases in a non-linear way. In particular, Figure 2c illustrates how the percentage of the successful transaction grows with the largest cluster size. It can be observed that the number of successful transactions steadily grows when the largest cluster is less than 900 nodes and then significantly increases. The fact that all clustering scenarios lead to higher number of successful transactions compared to the case with no clustering (which is intuitive and expected) illustrates how much our approach would improve upon the approaches based solely on credit networks. At this point we emphasize that the clusters (either small or large) are not meant to be predefined in advance. The nodes should instead

be incentivized by proper mechanisms to join clusters and thus contribute to their own and to the global success. In Figure 2b we then consider equally sized clusters, i.e., we have two and then three large clusters. These scenarios correspond to a situation where two or three large groups act as allies, where individuals within the group trust each other, however groups themselves are antagonistic to each other. Although the clustering again provides higher success rate it is to a lesser extent than in the previous case.

2) *Impact of reputation distribution:* We performed the next experiment in order to observe how different reputation distributions affect the final outcome in terms of transaction success. We compared three experiments where the reputation scores were selected as follows. In the first case all the reputation values were equal (in our case equal to 1). In the second case they were selected uniformly at random from the

interval from 0 to 10. In the third case they were selected according to a power-law distributed variable whose values are selected from the interval between 0 and 100. Since we calculate the currency values depending on the reputations directly, in the last two cases we have uniformly and power-law distributed currency values, as well. Figure 2d illustrates the number of successful transactions for this case.

We observe that the cases where reputations and consequently currency values are either equal or selected uniformly at random have almost the same performance. On the other hand in the third scenario, which is a model of network with huge differences in currency values, the performance is significantly lower. This indicates that a proper incentive mechanism is needed that will motivate the nodes who offer poor quality to strive for a better quality. Our Gesell economy inspired model provides the right incentive for this, since the nodes' individual account values depend directly on its quality of service and reputation. Figure 2e illustrates the values of individual accounts. The values of individual accounts are calculated as products of the total amount of money/currency multiplied by the value of that currency. It is obvious that the distribution of values of money nodes make corresponds very well to the same power-law distribution of nodes' reputations. Thus by trying to increase its own welfare by improving its reputation, a node directly contributes to the economic welfare of the community.

3) *Impact of the size of the graph:* In this experiment we varied the graph size. We set the number of nodes to be 256, 512, 1024, 2048 and 4096 respectively and selected the largest cluster size to be approximately 90% of the network's size. We observed that the percentage of successful transactions was always between 70% and 80% which indicates a good scalability. In particular for the network size of 256 nodes we obtained 76.00%, for the network size of 512 nodes we obtained 72.86%, for the size of 1024 nodes we obtained 81.70%, for the size 2048 we got 78.70% and for the largest size of 4096 we tested, we got 70.65% success rate. We emphasize at this point that we do not impose the size of the largest cluster to be always 90% of the network's size. This parameter was set to a fixed value in order to test the scalability of the model with different population sizes. In reality the users should make individual decisions regarding which clusters to join.

4) *Impact of the density of the graph:* Next we observed how the density of the underlying social graph affects the transaction success. For this purpose we varied the parameters a and b specific for the power-law node degree distribution, the maximal node degree $maxdeg$ and the clustering coefficient exponent. The results are given in Figure 2f and Figure 2g. The initial values for these parameters were: $a = 0.08$ and $b = 0.3$, $maxdeg = 64$ and clustering coefficient exponent 0.37. We have changed these values by increasing and decreasing them a little bit, since we did not want to perturb the original social graph too much. We observe that even small changes in these parameters affect the success rate. Decreasing the clustering coefficient exponent leads to worse success rate and increasing

it improves the rate. Thus the higher the density of the graph is the better the success rate. Increasing the values of a and b means more skewed distribution and a little decrease in the success rate. The maximal degree value does not appear to have a significant influence.

5) *Impact of account limits:* Finally we changed the account limits for every node to observe its impact and plot the results in Figure 2h. It turns (as expected) the account limits have huge influence on the success rate. We can observe that when we increase the credit limits by a factor of two and three respectively, after 1000 transactions the success rate increases by almost the same factors, i.e., two and three.

C. Implementation in CometCloud

Using the outcome of simulation, our implementation (currently in progress) makes use of the CometCloud framework³. Conceptually, CometCloud is composed of a programming layer, service layer and an infrastructure layer. The infrastructure layer uses the Chord self-organizing overlay [12], and the Squid [13] information discovery and content-based routing substrate that is built on top of Chord. The routing engine supports flexible content-based routing and complex querying using partial keywords, wildcards and ranges. It also guarantees that all peer nodes with data elements that match a query/message will be located. The service layer supports a Linda-like [14] tuple space coordination model, and provides a virtual shared-space abstraction as well as associative access primitives. Dynamically constructed transient spaces are also supported and enable applications to explicitly exploit context locality to improve system performance. In this paper, we build the proposed social cloud community model using such a shared space. The programming layer provides the basic framework for application development and management. It supports a range of paradigms including the master/worker/Bag-Of-Tasks. Masters and workers can communicate via virtual shared space or using a direct connection. Scheduling and monitoring of tasks are supported by the application framework. The task consistency service handles lost/failed tasks. Other supported paradigms include workflow-based applications as well as MapReduce/Hadoop. We propose two implementation designs based on master/worker: i) one tuple space per cluster/community; (ii) one coordination tuple space and multiple transient spaces – one for each cluster/community. CometCloud uses a pull-based tuple consumption model: master nodes generate transactions between peers and insert them (in the form of tuples) into the tuple space, then the workers pick up tasks from the tuple space and consume them. Workers model both provider and consumer nodes of the Social Cloud. These (secure) workers can access and possibly host part of the virtual shared space coordination infrastructure; however, isolated workers (can get tuples only through a proxy, a gateway node to access the shared space) can be eventually used for scalability/security reasons.

³<http://www.cometcloud.org>

IV. RELATED WORK

Within an economic market that operates within a *local* community, participants can generate their own currency by using a valuable service as an exchangeable object. Local, in this context, implies that the participants are known to each other, and therefore utilize pre-existing trust relationships. Several experiments, such as Wärgl in 1932 (stamp scrip [15]), Comox Valley in 1983 (LETS: Local Exchange Trading System [16]) and in Ithaca since 1991 (Ithaca Hours [17]) used the barter of complementary currencies for supporting values such as volunteer work, daily help or skills that are often not regularly utilized within a financial market.

Facilitating the exchange of computational and data services by using complementary currencies has become an alternative solution for Peer-2-Peer (P2P) networks as powerful instruments to promote exchange. iWAT [18], Geek Credit [19] and Ripple [20] are examples of systems that use such complementary currencies for exchanges of computational service such as storage, computational capacity and network bandwidth. Within such a market, a peer node can exchange capability that it does not immediately need for an alternative service – thereby enabling a better sharing of overall capacity between participants in the market.

iWAT [18] is an internet version of the WAT system and implements the core aspects of WAT – focusing specifically on the problem of ticket exchange. As an issued ticket can circulate between a number of different users within the WAT system, the party responsible for issuing the ticket is also responsible for validating its authenticity – as the redeemer of the ticket may be different from the one to whom the ticket was initially issued. The system also provides incentive mechanisms for keeping users online, thereby providing a large enough population of users to enable ticket exchange.

Samsara [21] concentrates on establishing service relationships between peers, minimising risk due to failure and claim, to ensure equivalence between the contribution and consumption of peer nodes. Establishing symmetric storage relationships leads to fairness and is handled as a central goal. Samsara is similar to the WAT system, however although designed as a mechanism to support exchanging objects for storage replication, it is preoccupied with fairness, rather than offering an accurate payment representation.

The Ripple protocol [20] enables trustworthy ‘IOU’ exchanges defining particular trusted connections for each currency. Each currency operates on a separate user network, but participants also have the authority to accede in different currency networks. It is assumed that one unit of a product has equal significance for all the participants. In most cases this assertion represents only an assumption; e.g. one commodity has one significance for A but probably has different significance (utility) for B.

V. CONCLUSION AND FUTURE WORK

We propose an incentive model for resource sharing in social clouds. Our model extends work in credit networks, does not require a central reputation management system and

supports asynchronous demands. In addition, we utilize trust inherent within social networks to achieve more flexibility by introducing a virtual currency representation motivated by work of Gesell. We study the long term liquidity, that is the capacity to route payments, when nodes repeatedly transact with each other. Using simulations we show how network size, clustering strategy, reputation distributions, density of the social graph and credit limits impact the success of transactions and the overall credit gain. Our future work will focus on studying other realistic scenarios such as network and cluster dynamics and non-cooperative behavior, along with deploying the model using the CometCloud framework.

REFERENCES

- [1] D. Neumann, C. Bodenstein, O. F. Rana, and R. Krishnaswamy, “STACEE: Enhancing Storage Clouds using Edge Devices,” in *Proceedings of the 1st ACM/IEEE workshop on Autonomic computing in economics (ACE), alongside ICAC 2011*, 2011, pp. 19–26.
- [2] K. Chard, K. Bubendorfer, S. Caton, and O. Rana, “Social Cloud Computing: A Vision for Socially Motivated Resource Sharing,” *IEEE Transactions on Services Computing*, no. 99, p. 1, 2011.
- [3] I. Petri, O. Rana, Y. Rezgui, and G. C. Silaghi, “Evaluating Trust in Peer-to-Peer Service Provider Communities,” in *Processings of IEEE CollaborateCom*, 2011, pp. 407–414.
- [4] S. Gesell, *The Natural Economic Order, Revised edition*. London: Peter Owen, 1958.
- [5] Z. Liu, H. Hu, Y. Liu, K. W. Ross, Y. Wang, and M. Mobius, “P2P Trading in Social Networks: The Value of Staying Connected,” in *INFOCOM*, 2010, pp. 2489–2497.
- [6] P. Dandekar, A. Goel, R. Govindan, and I. Post, “Liquidity in Credit Networks: a Little Trust goes a Long Way,” in *ACM Conference on Electronic Commerce*, 2011, pp. 147–156.
- [7] D. B. DeFigueiredo and E. T. Barr, “TrustDavis: A Non-Exploitable Online Reputation System,” in *CEC*, 2005, pp. 274–283.
- [8] A. Ghosh, M. Mahdian, D. M. Reeves, D. M. Pennock, and R. Fugger, “Mechanism Design on Trust Networks,” in *WINE*, 2007, pp. 257–268.
- [9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The Eigentrust Algorithm for Reputation Management in P2P Networks,” in *WWW ’03*, 2003, pp. 640–651.
- [10] J. Leskovec and E. Horvitz, “Planetary-scale Views on a Large Instant-messaging Network,” in *WWW ’08*, 2008, pp. 915–924.
- [11] E. Ravasz and A. L. Barabasi, “Hierarchical Organization in Complex Networks,” *Phys Rev E*, vol. 67, no. 2, 2003.
- [12] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a Scalable Peer-to-Peer Lookup Protocol for Internet Applications,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.
- [13] C. Schmidt and M. Parashar, “Squid: Enabling search in DHT-based systems,” *J. Parallel Distrib. Comput.*, vol. 68, no. 7, pp. 962–975, 2008.
- [14] N. Carriero and D. Gelernter, “Linda in Context,” *Commun. ACM*, vol. 32, no. 4, 1989.
- [15] F. Schwarz, “Das experiment von wärgl,” 1951, shortened English translation by Hans Eisenkolb is available at <http://www.sunshinecable.com/~eisehan/woergl.htm>. [Online]. Available: <http://userpage.fu-berlin.de/roehrigw/woergl/>
- [16] S. Seron, “Local Exchange Trading Systems 1 - Creation and growth of LETS,” [Online]. Available: <http://www.gmlts.u-net.com/resources/sidonie/home.html>
- [17] P. Glover, Ithaca Hours. [Online]. Available: <http://www.ithacahours.com>
- [18] K. Saito, “i-WAT: The Internet WAT System – An Architecture for Maintaining Trust and Facilitating Peer-to-Peer barter Relations,” Ph.D. dissertation, School of Media and Governance, Keio University, 2006.
- [19] A. Komarov, Geek Credit. [Online]. Available: <http://home.gna.org/geekcredit/>
- [20] R. Fugger, “Money as IOUs in Social Trust Networks and a Proposal for a Decentralized Currency Network Protocol,” 2004. [Online]. Available: <http://ripple.sourceforge.net/>
- [21] L. P. Cox and B. D. Noble, “Samsara: Honor Among Thieves in Peer-to-Peer Storage,” *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 120–132, 2003.