Social Cloud Computing: A Vision for Socially Motivated Resource Sharing

Kyle Chard, *Member*, *IEEE*, Kris Bubendorfer, *Member*, *IEEE*,
Simon Caton, *Member*, *IEEE*, and Omer F. Rana, *Member*, *IEEE*

Abstract—Online relationships in social networks are often based on real world relationships and can therefore be used to infer a level of trust between users. We propose leveraging these relationships to form a dynamic "Social Cloud," thereby enabling users to share heterogeneous resources within the context of a social network. In addition, the inherent socially corrective mechanisms (incentives, disincentives) can be used to enable a cloud-based framework for long term sharing with lower privacy concerns and security overheads than are present in traditional cloud environments. Due to the unique nature of the Social Cloud, a social market place is proposed as a means of regulating sharing. The social market is novel, as it uses both social and economic protocols to facilitate trading. This paper defines Social Cloud computing, outlining various aspects of Social Clouds, and demonstrates the approach using a social storage cloud implementation in Facebook.

Index Terms—Social Cloud, social networks, cloud computing, services computing

1 Introduction

Digital relationships between individuals are becoming as important as their real world counterparts. For many people social networks provide a primary means of communication between friends, family, and coworkers. The increasing ubiquity of social network platforms is evidenced by their rapid and ongoing growth. For instance, Facebook has over 500 million active users of which 50 percent log on every day.[1]

Users are more likely to *trust* information from a "friend" if the digital relationship between the two is based on a real world relationship (friend, family, colleague) rather than a purely online relationship (second life, online games, etc.). As relationships within online social networks are at least partly based on real-world relationships, we can therefore use them to infer a level of trust that underpins and transcends the online community in which they exist.

This implicit trust along with the application of socially corrective mechanisms (incentives, disincentives) inherent in social networks can also be applied to other domains. In fact, social networking platforms already provide a multitude of integrated applications that deliver particular functionality to users, and more significantly, social network credentials provide authentication in many diverse domains, for example, many sites support Facebook Connect as a trusted authentication mechanism.

Like any community, individual users of a social network are bound by finite capacity and limited capabilities. In many cases however, other members (friends) may have surplus capacity or capabilities that, if shared, could be used to meet fluctuating demand. A Social Cloud leverages preexisting trust relationships between users to enable mutually beneficial sharing within the context of a social network. It is important to note that sharing within a Social Cloud is not representative of point-to-point exchanges between users, rather it represents multipoint sharing within a whole community group. We now define a Social Cloud explicitly as:

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

The resources exchanged need not be symmetric and can represent vastly different capabilities. A cloud-based usage model is used to enable virtualized resource sharing through service-based interfaces.

Fig. 1 shows the different aspects of the Social Cloud model that are explored in this paper. In Section 2, we discuss the different individual and group relationships represented in social networks, the basis of trust and social incentives on which the Social Cloud is anchored, and the social market that regulates sharing in a Social Cloud.

We present a range of Social Cloud application scenarios in Section 3, followed by our architecture for, and implementation of, the social storage cloud scenario in Section 4. An empirical evaluation of the social storage cloud is presented in Section 5, followed by a reflective

1. http://www.facebook.com/press/info.php?statistics, last accessed April 2011.

• *K. Chard is with the Computation Institute, University of Chicago, 5735 South Ellis Avenue, Chicago, IL 60637 and Argonne National Laboratory, Chicago, IL. E-mail: kyle@ci.uchicago.edu.*
• *K. Bubendorfer is with the School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand. E-mail: kris.bubendorfer@ecs.vuw.ac.nz.*
• *S. Caton is with the Karlsruhe Services Research Institute and the Institute of Information Systems and Management, Information & Market Engineering (IM), Karlsruhe Institute of Technology, Englerstr. 14, 76131 Karlsruhe, Germany. E-mail: simon.caton@kit.edu.*
• *O.F. Rana is with the School of Computer Science and Informatics, Cardiff University, Queen's Buildings, Newport Road, Cardiff CF24 3AA, United Kingdom. E-mail: o.f.rana@cs.cardiff.ac.uk.*

Manuscript received 29 Nov. 2010; revised 24 Apr. 2011; accepted 2 June 2011; published online 16 June 2011.
For information on obtaining reprints of this article, please send e-mail to: tsc@computer.org, and reference IEEECS Log Number TSCSI-2010-11-0148.
Digital Object Identifier no. 10.1109/TSC.2011.39.

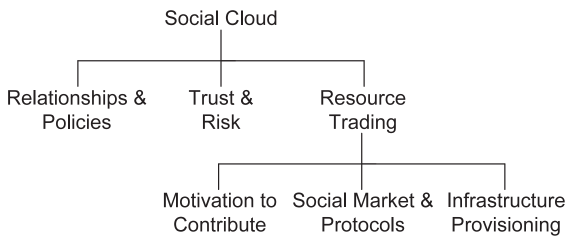

Fig. 1. Aspects of a Social Cloud.

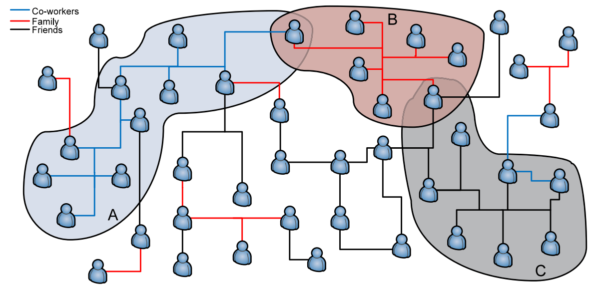

Fig. 2. Social Cloud overlay in a social network. Three different Social Clouds are illustrated to highlight the use of relationships when establishing Social Clouds.

analysis of the social storage cloud against our design goals in Section 6. A survey of related work is presented in Section 7. Finally, we present future work in Section 8, and our concluding remarks in Section 9.

2 Social Cloud Computing

The act of adding an individual as a social network "friend" implies that a user has some degree of knowledge of the individual being added. Such connectivity between individuals can be used to infer that a trust relationship exists between them. However, it does not describe the level of trust or the context of the relationship. For instance a "friend" can be a member of the family, a work colleague, a college affiliate, a member of the same sports club, etc.

Facebook has recently recognized the need for the creation of such groups and allows users to differentiate between, for example, close friends and colleagues. In a Social Cloud, this provides the basis for defining different levels of trust based on the group abstraction supported by the infrastructure. For example, a user could limit sharing with close friends only, friends in the same country, network or group, all friends, or even friends of friends.

Another way of thinking about the Social Cloud is to consider that social network groups are analogous to dynamic Virtual Organizations (VOs) [1]. Groups, like VOs, have policies that define the intent of the group, the membership of the group, and sharing policies for the group. This model is illustrated in Fig. 2, where user-specific groups, defined by relationship types, are shown in the context of a social network. In this example group A is composed of only coworker members, whereas group B is formed by family members and group C includes only friends. Clearly the level of trust and mechanisms for *social correction* (identifying incentives and disincentives for users to participate) differ between groups. This figure also highlights that Social Clouds are not mutually exclusive, that is, users may be simultaneously members of multiple Social Clouds. Whereas a VO is often associated with a particular application or activity, and is often disbanded once this activity completes, a group is longer lasting and may be used in the context of multiple applications or activities. We take this latter view, and use the formation of social groups to support multiple activities.

In addition, different sharing policies and market metaphors can be defined depending on the group, for instance a user may be more likely to share resources openly with family members without requiring a high degree of reciprocation, however the same might not be true for work colleagues or scientific collaborators.

2.1 Trust and Risk

Commercial cloud providers typically offer few explicit guarantees, instead they rely on implied trust based on the commercial standing of the provider. For the individuals sharing resources within a Social Cloud this approach is not feasible, and therefore it is important to use social incentives and the underlying *real world* relationships as a substitute foundation for trust. At present, none of the major social networks are able to provide guarantees about the real-world identity associated with a user profile. To do so, explicit identification processes, such as those used in Safebook [2], are required to ensure profiles are mapped to a real person or organization.

Social correction through incentives encourages *good behavior* without external enforcement. A Social Cloud must leverage social incentives to create ad hoc clouds without incurring the overhead of complex enforcement processes present in commercial environments. This approach can also be used to overcome one of the major limitations of cloud computing, i.e., the creation, monitoring, and enforcement of Service Level Agreements (SLAs). In a social context formal SLAs are not as critical because individuals are socially motivated and subject to personal repercussions outside the functional scope of the Social Cloud.

In addition to trust, the level of risk must also be considered within a Social Cloud. Take, for example, a storage service, consumers are risking loss, unavailability, compromise, or corruption of data while providers are risking their own environment and exploitation of their resources by hosting this data. In a Social Cloud users may want to minimize these risks. In the storage scenario, providers can alleviate risk through service design and sandboxing, while consumers can avoid compromising file content through encryption, or reduce the impact of file loss through replication.

2.2 Resource Trading

A Social Cloud resource represents a physical or virtual entity (or capability) of limited availability. A resource could therefore encompass people, information, computing capacity, or software licenses—hence, a resource provides a particular capability that is of use to other members of a group or community. Resources shared in a Social Cloud are by definition heterogeneous and potentially complementary, for example, one user may share storage in exchange for access to a specific workflow. Or in another example, a user may back up photos from their digital camera to the hard disk of another member in the social network. To participate in a Social Cloud, each user must allocate a certain amount of their resources to be used by others. The sharing is controlled (or regulated) by a socially oriented market place which adapts common allocation protocols to a social context.

2.2.1 Motivation for Contribution

The underlying social incentives present in a Social Cloud motivate users to participate in, and contribute to, their community in different ways. Motivation has been studied in a number of other online domains [3], for example, sharing information and photos in social networks, sharing metadata and tags in online communities, and collaborative knowledge building through online content projects (e.g., Wikipedia) or open source software projects. Motivation is generally categorized as either intrinsic or extrinsic.

Extrinsic motivation represents the case where users are motivated by an external reward (e.g., money), they will therefore contribute to a community while the expected benefits exceed the cost of contribution even when they have little interest in the community. Examples of extrinsic motivation include: improvement of skills [4] (learning from others and receiving feedback), enhancement of reputation through contribution [5] (brand awareness, professional status), and traditional financial motivation.

Intrinsic motivation represents an internal satisfaction obtained from the task itself rather than the rewards or benefits. This sense of satisfaction may be from completing a task or even simply working on it. Examples of intrinsic motivation include: enjoyment [6], reciprocation [7], commitment to a cause [8] (obligation to contribute), and tenure in the community [3].

These motivating factors rationalize noneconomic contribution to existing online communities. Each factor also motivates users to contribute to and use a Social Cloud.

2.2.2 Compensation and Fairness

Compensation in the general sense is optional in a Social Cloud as users may wish to share resources without payment, and rather utilize a reciprocal credit (or barter)-based model to increase "social capital." Alternatively users may share excess capacity to generate real revenue (whereby each "credit" gained maps to a real currency—as in Facebook) or social revenue where the cumulative contribution to the network opens up new sharing opportunities. This model is somewhat similar to a volunteer computing approach, in that friends share resources for little or no gain. However, unlike volunteer models there is inherent accountability through existing friend relationships.

2.2.3 Social Capital

Social capital represents an investment in social relationships with expected returns [9]. From an individual standpoint, social capital is similar to human capital in that users of a social network may gain individual returns for specific actions (e.g., selling goods or finding a new job). From a group perspective, social capital represents the intrinsic (intangible) value of the social community, that is, the community as a whole generates returns by the actions (or cooperation) of its members.

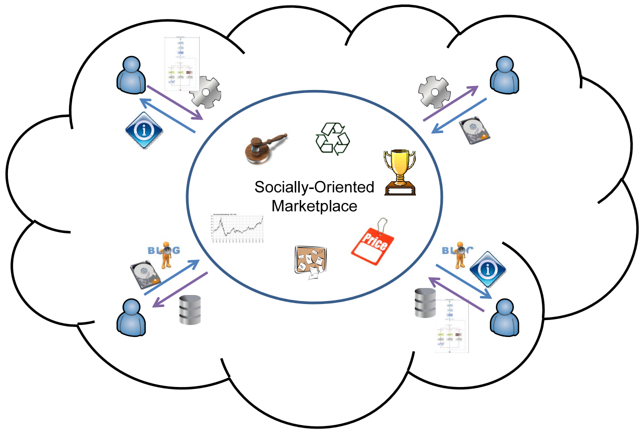

Fig. 3. Capability sharing in a Social Cloud. Users contribute resources or capabilities in exchange for asymmetric resources contributed by "friends."

With the growth of online relationships there is potential to create new forms of social capital due to the ease with which online social networks allow users to create and maintain large, distributed networks of relationships [10], [11]. The sharing of resources in a Social Cloud is one such opportunity to invest and generate value from the actions of individuals. In fact, the sharing model in a Social Cloud could be considered as generating both social and physical capital as it is reflective of the real world—the resources shared (compute, storage) have been invested in by their owners and are expected to produce some individual return. Sharing such resources in a social context can therefore benefit both the community and the investor. In a Social Cloud individuals invest in the community by joining the cloud, sharing resources and utilizing other's resources. The sharing is, in effect, an investment in the community and its members, the result of which may be very diverse.

2.2.4 The Social Market

The *Social Marketplace* is at the core of the Social Cloud and is used to regulate sharing within a group. Each group is associated with a separate instance of the market. The market is shown in Fig. 3. The marketplace is tasked with allocating resources between peers according to predefined economic or noneconomic protocols. Traditionally, a marketplace is assumed to be based on the exchange of goods for money, however in a Social Cloud the marketplace is not necessarily monetary. For example, a nontangible trophy system (similar to that used in volunteer models) may provide suitable incentives to encourage sharing among friends, in this case the marketplace is responsible for managing the trophy model and regulating exchange. In general, the Social Marketplace is defined by the needs of the group and in many cases different types of market metaphors may coexist.

2.2.5 Social Market Metaphors and Protocols

A Social Marketplace contains a set of market protocols tasked with determining the most appropriate allocation given to a particular user request. The choice of protocol is dependent on the Social Cloud and the requirements of its members. Examples of common protocols include:

- **Volunteer.** An idealistic sharing model in which users contribute resources for no personal gain—but do so without accountability for their actions [12], [13], [14].
- **Trophy.** A nonmonetary model in which users are rewarded with nontangible credits or prizes (fame) for achieving contribution goals [12], [15], [16]. Trophy systems have been successfully used as an add-on by Volunteer computing projects as a means of encouraging participation.
- **Reciprocation.** A sharing model in which users that contribute the most to the cloud are proportionally favored when requesting resources [17], [18], [19].
- **Reputation.** A model based entirely on a measure of individual reputation [20], [21], [22], [23]. Reputation is established through interactions in the community (which may not necessarily be within the Social Cloud), when allocating resources those with higher reputation are favored.
- **Posted price.** A model in which market resources are offered at a set (posted) price [24], [25]. A posted price model is the predominant economic model employed by commercial cloud providers.
- **Auction/tender.** A dynamic multiparticipant mechanism designed to establish the market price for a particular resource [26], [27], [28], [29]. Auctions are used extensively for online sales of goods through sites such as eBay.
- **Spot price.** A dynamic pricing protocol in which a commodity is offered at a price given at a particular time and location [30], [31]. Amazon EC2 offers a competitive hybrid Spot price market[2] to facilitate dynamic pricing, if the bid is greater than the current spot price the instance is provisioned.

2.2.6 Provision of the Trading Infrastructure

The host infrastructure for a Social Cloud could be provisioned in multiple ways, for example, it could be provided externally (i.e., outsourced to an external vendor) or internally by the members themselves. Using an external provider is potentially easier, however it may be expensive and might not scale if a single market instance vendor is used for all groups. Supplying the infrastructure internally can more easily scale with the size of the group and it maps to the philosophy of social contribution inherent in a Social Cloud, however it requires a high degree of trust and cooperation between users.

A *co-op* is a business owned and operated by a community for the mutual benefit of the community (e.g., community managed grocery stores and credit unions). In a co-op customers using the business are also partial stake holders in the business, they therefore own the property, employ the workers, and have input into how the business is managed. A similar model is particularly apt when considering a Social Cloud. Due to the inherent social incentives the market itself can be hosted on resources contributed by members of the community, therefore establishing a Social Cloud co-op. This alleviates the expense involved in outsourcing the infrastructure and it can also scale with the size of the community as all members contribute infrastructure to the cloud.

The limitation with a co-op is the potential for malicious behavior. In a noncommercial environment social incentives may be sufficient to ensure nonmalicious behavior, however in a competitive economy with considerable resources available these incentives may not guarantee the behavior of the market and its participants. These problems are amplified when considering that the market could be hosted on potentially untrusted hosts. The major issues are guaranteeing that the allocation is carried out fairly (not subverted) and that private information is not disclosed (e.g., pricing data). There are various ways to establish trust in a particular market service operating in an untrusted environment [32]: for example, using reputation, encryption, or threshold trust. One of the best techniques is the use of secure economic protocols [33], [34], [35] which, through encryption and distribution are able to provide guarantees over market execution and economic privacy. Such protocols allow a market to be safely executed on participating hosts without revealing information, there is however, a tradeoff between market security and (computational) efficiency.

3 Application Scenarios

The potential application scenarios that benefit from cloud models are immense (from scalable web servers through to data intensive scientific applications). The point of difference of a Social Cloud is that applications can also leverage the relationships between users to deliver shared asymmetric services—leading to several potential Social Cloud application scenarios:

- **A social computation cloud.** It is widely recognized that extensive computing power remains untapped through personal computers. The use of a Social Cloud provides an infrastructure from which users can easily contribute computing resources to friends, companies, or scientific communities (similar to a volunteer computing project).
- **A social storage cloud.** Storage is perhaps the simplest and most standardized resource for everyday users to share and utilize in a Social Cloud. Online data storage is commonly used to store, backup, share, and replicate data. One obvious use for social storage is storing and sharing photos. While most social networks already store photos, the burden for hosting them could be moved from the network provider to their members to increase scalability and reduce infrastructural requirements. The security implications are limited as photos are typically already shared with friends.
- **A social collaborative cloud.** Increasingly, collaborations are turning to social networking concepts to share information and resources within diverse user communities, for example, MyExperiment.org [36] and nanoHUB.org [37]. Similar functionality can be realized using dynamic Social Clouds deployed in existing social networks. Storage services can be used to store/share data and information (for example, academic papers, scientific workflows, data sets, and analysis) while computation (or specific scientific

2. http://aws.amazon.com/ec2/spot-instances, last accessed April 2011.

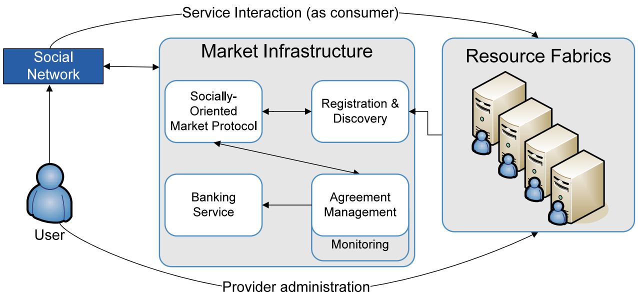

Fig. 4. Social Cloud architecture. Users register shared services, their friends are then able to provision and use these resources through the social storage cloud application. Allocation is conducted by the underlying market infrastructure(s).

services such as workflows) can be used to execute scientific applications. A Social Cloud approach is advantageous as there is no requirement for dedicated infrastructure or management, fewer barriers to entry for new communities, and users can utilize existing social network accounts.

- **A Social Cloud for public science.** The Social Cloud is an ideal basis on which to create the next iteration of volunteer computing—primarily for solving scientific problems of community interest. There are many examples of such projects run as volunteer computing problems under the Berkeley Open Infrastructure for Network Computing (BOINC) [13], such as SETI, Rosetta, Docking, etc., and these projects have been able to leverage massive computing power from donated resources. A Social Cloud for science can do this in a more accessible way and leverage a larger population base, using different resource provider groups to determine share delegation, finer grained resource control, the integration of social capital, reputation, and social incentives. This approach can lower the entry barriers to the donation or (temporal) trading of computing power, and can indeed be utilized for more highly cooperative structures between research groups, small organizations, and forge mutually constructive scientific communities.
- **An enterprise Social Cloud.** A Social Cloud may be configured differently, depending on the community it serves. It is increasingly common for organizations to have a social network presence, for example, many companies, universities, and schools all have public social network profiles. This presents an opportunity for large scale users to form specialist enterprise Social Clouds. From a provider's perspective, the benefits are twofold: not only do they gain access to a pool of resources when required they may also benefit from the social rewards of sharing—for example, enhancing brand awareness and increasing public perception of the organization.

4 The Social Storage Cloud

To demonstrate the feasibility of the Social Cloud, a web service-based social storage cloud has been developed and deployed as a Facebook application. In the social storage cloud, two economic markets have been created; both operate independently and are designed to work simultaneously. In a posted price market, users select storage from a list of friends' service offers. In the reverse auction (tender) market, consumers outline specific storage requirements and pass this description to the Social Cloud infrastructure; providers then bid to host the storage. Both markets result in the establishment of an SLA between users. The SLA is redeemed through the appropriate storage service to create a storage instance. In such a social market, participating users know the corresponding user's identity and can directly interact with the provider to identify why a particular capability was not delivered. However, where no such prior relationships exist, an SLA provides a more appropriate mechanism, requiring reward and penalty clauses to limit risk for the user and liability for the provider. However, the use of an SLA still remains a useful capability to support within a social storage cloud, as this could be subsequently generalized to deal with varying types of trust relationships between individuals involved as users or providers of resources.

The general architecture of the social storage cloud is shown in Fig. 4. The social network provides user and group management as well as the medium to interact with the Social Cloud infrastructure through service-based resource interfaces. The core market infrastructure is responsible for facilitating sharing and includes components for capability registration and discovery, implementing and abstracting the chosen market protocol, managing and monitoring provisions, and regulating the economy.

4.1 Facebook Applications

Facebook exposes access to their social graph through the OpenGraph API,[3] through the Representational State Transfer (REST) service interface applications can access all objects (friends, events, groups, application users, profile information, and photos) and the connections between them. To access the OpenGraph API both the user and the application must be authenticated, in Facebook this process uses the OAuth protocol [38]. The Facebook user and application authorization model was one of the motivating factors for choosing Facebook.

Facebook Markup Language (FBML) includes a subset of HTML with proprietary extensions that enables the creation of applications that integrate completely with the Facebook look and feel. Facebook JavaScript (FBJS) is Facebook's version of JavaScript—rather than sandboxing JavaScript, FBJS is parsed when a page is loaded to create a virtual application scope. All Facebook applications are hosted independently by application providers. A Facebook canvas URL is created for user access, this URL maps to a user defined callback URL which is hosted remotely. The process of rendering an application page is shown in Fig. 5. When a page is requested by the user through the Facebook canvas URL the Facebook server forwards the request to the defined callback URL. The application creates a page based on the request and returns it to Facebook. At this point the page is parsed and Facebook specific content is added according to the FBML page instructions. The final page is then returned to the user. This routing structure presents an

3. http://developers.facebook.com/docs/api, last accessed April 2011.

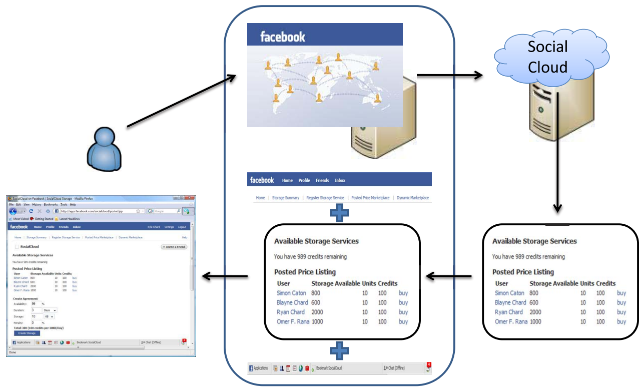

Fig. 5. Facebook application hosting environment. The Social Cloud web application generates page content which is parsed by Facebook to create the page delivered to the user.

important design consideration for a Social Cloud as access to the cloud services would be expensive if routed through both the Facebook server and the callback application server in order to get data from the actual cloud service. To reduce this effect, FBJS can be used to request data asynchronously from the specified service in a transparent manner without routing through the application server.

4.2 Storage as a Service

There are two generic requirements of the shared storage service: first, the interface needs to provide a mechanism to create a stateful instance for a reservation. In our model, the social storage cloud application passes a WS-Agreement [39]-based SLA to the service which is parsed and used to instantiate the required state. Second, in order to be discovered the service needs to advertise capacity so that it can be included in the market. In the social storage cloud this advertised capacity is encoded using XML-based metadata which is periodically refreshed and stored in a Globus Monitoring and Discovery System (MDS) [40]. The social storage cloud is based on a generic Web Services Resource Framework [41] (WSRF) storage service, which provides an interface for users to access virtualized storage. This service exposes a set of file manipulation operations to users and maps their actions to operations on the local file system. Users create storage by passing an agreement to the storage service, this creates a mapping between a user, an agreement, and the storage instance. Instances are identified by a user and an agreement allowing individual users to have multiple storage instances in the same storage service. The storage service creates a representative WSRF resource and an associated working directory for each instance. The resource keeps track of service levels as outlined in the agreement such as the data storage limit. Additionally, the service has interfaces to list storage contents, retrieve the amount of storage used/available, upload, download, preview, and delete files.

Each storage service relies on a web application to deliver content to the Facebook application without routing data through the Social Cloud application. To do this the storage service has a REST API that performs a specific action (corresponding to the service) and delivers a response in the form of JavaScript Object Notation (JSON). This approach allows dynamic Asynchronous Javascript and XML (AJAX) invocation of storage operations without requiring a callback or page reload of the Social Cloud application.

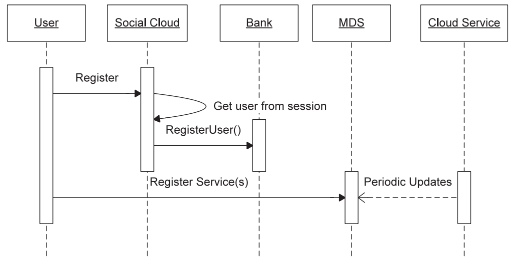

Fig. 6. Registration in a Social Cloud. For brevity, the diagram assumes the user has been authenticated.

4.3 Currency Regulation

The social storage cloud includes a credit-based system that rewards users for contributing resources and charges users for consuming resources. A banking service registers every member of the cloud by storing their credit balance and all agreements they are participating (or have participated) in. Credits are exchanged between users when an agreement is made, prior to the service being used. To bootstrap participation in the Social Cloud, users are given an initial number of credits when joining the cloud. While suitable for testing, this initial credit policy is susceptible to inflation and cheating (if fake users are created and the initial credits are transferred). Currently, there is no mapping between Social Cloud credits and real currencies or Facebook credits.

The banking service is composed of two associated context services each representing different instance data. The first context service manages user resources while the second manages storage agreements. The user resource stores the user's Facebook ID, current credits, agreement IDs the user has participated in, and auction references. The agreement resource contains any agreements created in the system, which is used to manage provision information and act as a receipt.

4.4 Registration

The registration process is shown in Fig. 6. Upon joining the cloud users first need to register themselves, and then specify the cloud services they are willing to trade. As users are preauthenticated through Facebook, user instances can be transparently created in the banking service using the user's Facebook ID. Having registered, the user is presented with an MDS EndPoint Reference (EPR) and cloud ID which they use to configure their cloud services for registration (and refreshment) of resource capacity. Market services utilize the MDS XPath interface to discover suitable services based on user IDs and real-time capacity. The service updates its metadata whenever resource state changes, this update is reflected in MDS according to the application policies.

4.5 Social Marketplace

The social storage cloud implementation includes two concurrent economic markets, posted price and reverse auctions.

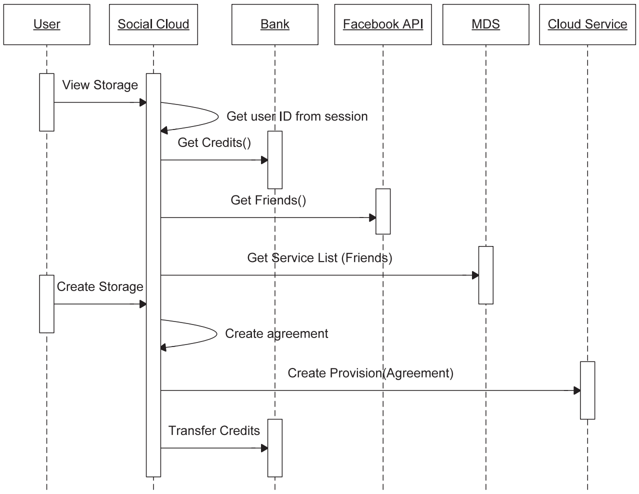

Fig. 7. Posted Price marketplace in a Social Cloud. For brevity, the diagram assumes the user has been authenticated.

4.5.1 Posted Price

In the posted price marketplace, a user can select any advertised service and define specific requirements (storage amount, duration, availability, and penalties) of the provision. Fig. 7 shows the flow of events for a posted price trade in the social storage cloud. After logging on, the social storage cloud application transparently validates the Facebook user ID through the banking service to ensure the user is registered and to also retrieve their current number of credits. A list of all the user's friends is generated using the Facebook REST API, this list is used to compose a query to discover friends' storage services from MDS. The result of this query is used to populate the posted price offer list that describes availability and pricing information. When the user selects a service they also specify their required service levels, an SLA is created using the SLA creation component of SORMA [42]. To do this, the storage requirements are encoded into an EJSDL [43]; (JSDL [44] with economic extensions) document describing the storage request. This document is then converted into an agreement using SORMA SLA tools. The EJSDL document acts as the Service Description Term of the agreement and individual requirements are split into guarantee terms (as defined in [43]). EJSDL extends JSDL by adding additional economic information describing pricing and penalties which are mapped to their respective Business Value Lists. We have further extended this term language to include two additional cloud specific quality of service (QoS) terms: Availability and Error Rate, which are defined as JSDL ranges and are used to describe and monitor the availability of the storage service. The duration of the service provision is encoded using the SORMA reservation specification.

Before service invocation, the generated SLA must be passed to the appropriate storage service to create a storage instance. The storage service determines if it will accept the agreement based on local policy and current resource capacity. Having instantiated the storage the agreement is then passed to the banking service to exchange credits and store a copy to act as a receipt. If either the banking service or storage service decline the agreement both entities remove the reservation.

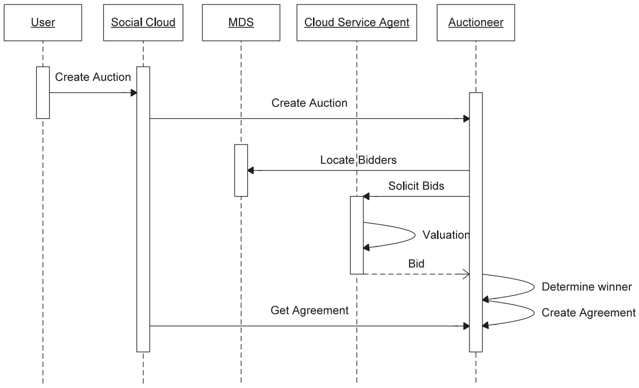

Fig. 8. Auction marketplace in a Social Cloud. For brevity, the diagram assumes the user has been authenticated and also excludes the actions taken to find the users' ID, retrieve the users' friends, instantiate the cloud service, and transfer credits—these actions are shown in Fig. 7.

4.5.2 Auctions

In the reverse auction (tender) market, a user can specify their storage requirements and then submit an auction request to the social storage cloud. The user's friends then bid to provide the requested storage. The auction mechanisms used are based on the DRIVE metascheduler [29]. In particular, a reverse Vickrey auction protocol plug-in is used, as the dominant bidding strategy (truth telling) is more socially centric. It also means that "antisocial" behavior such as counter speculation is fruitless.

Fig. 8 illustrates the auction process. In a reverse auction, cloud services compete (bid) for the right to host the user's task. The auctioneer uses the list of Facebook friends to locate a group of suitable storage services based on user specified requirements; these are termed the bidders in the auction. Each bidder then computes a bid based on the requirements expressed by the consumer. The auctioneer determines the auction winner and creates an SLA between the auction initiator and the winning bidder. As in the posted price mechanism, the agreement is sent to the specified service for instantiation and the bank for credit transfer.

In DRIVE, an Auction Manager (AM) is responsible for creating the auction, soliciting bids, and determining a winner. Individual Bidding Agents (BA) act on behalf of a user to compute valuations according to local policy and valuation functions. The standard DRIVE BA has been modified to interact with the Storage Service to check capacity and compute a (linear) bid based on current capacity. The Agreement Manager (AgM) is used to create a WS-Agreement-based SLA between the user and auction winner.

5 Evaluation

This section outlines measurements obtained from the deployed social storage cloud. The following experiments focus on the scalability and performance of the two social marketplaces and the feasibility of the proposed co-op infrastructure (as discussed in Section 2.2.6). For the following experiments it is assumed an average Facebook user has 130 friends.[4] The market-based experiments are run

4. http://facebook.com/press/info.php?statistics, last accessed April 2011.

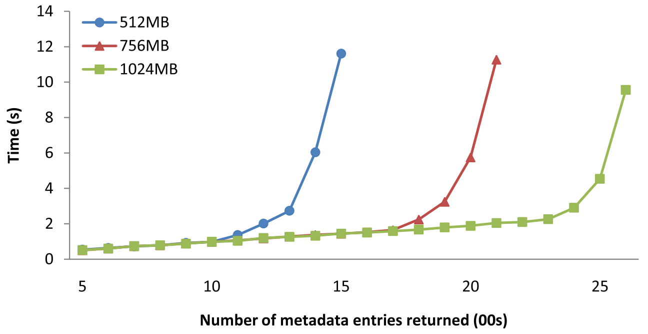

Fig. 9. Time taken to retrieve service metadata from MDS with different amounts of container memory.

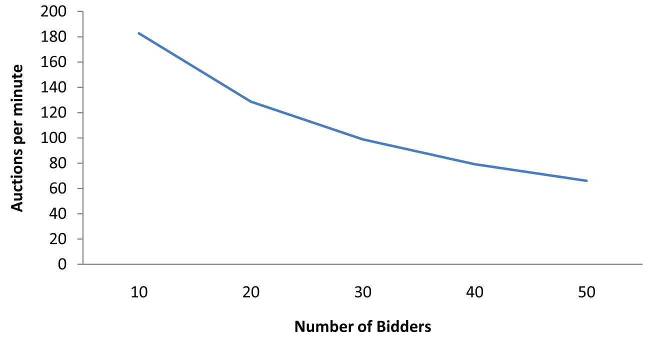

Fig. 11. Auction throughput. Number of auctions completed per minute for an increasing number of bidders.

on a single server running Windows Vista with a 2.2 GHz Dual Core processor and 2 GB memory. Bidders are hosted in a virtualized environment containing 5, 3.0 GHz Core 2 Duo machines each with 4 GB RAM.

5.1 Posted Price Allocation

Posted price trading requires several steps: identification of storage requirements, generation of an SLA, instantiation of a storage service, and registration of the transaction with the banking service. The time taken to perform these operations is constant and generally small compared to the time taken to discover storage offers, which is dependent on the MDS service.

Fig. 9 shows the time taken to query MDS for an increasing number of registered entries. The time includes the cost of converting the XML result into a Java object. Registration performance is shown to be dependent on the amount of memory given to the container and the number of registered entries. With 1 GB of memory over 2,000 offers can be retrieved in less than 2 seconds. Therefore, MDS can be run even on a low specification server yet still support a Social Cloud and its market.

In a Social Cloud policies dictate the services with which a user is willing to interact (e.g., friends, friends of friends). Such services can be identified by querying for registered services matching particular user IDs. Fig. 10 reflects this situation by loading an increasing number of services in MDS and querying for a subset of registered offers (friend's services). The query result ranges between 20 and 200 services, while the number of registered services is increased from 200 to 2,000. The container is running with 1 GB of memory. The time taken to retrieve entries is proportional to the number of registered services and also the number of services returned in the query. Assuming on average 130 friends per user, and the fact not all of these friends would be involved in a Social Cloud, this performance is acceptable—selecting 100 of 2,000 registered entries takes approximately 2 seconds.

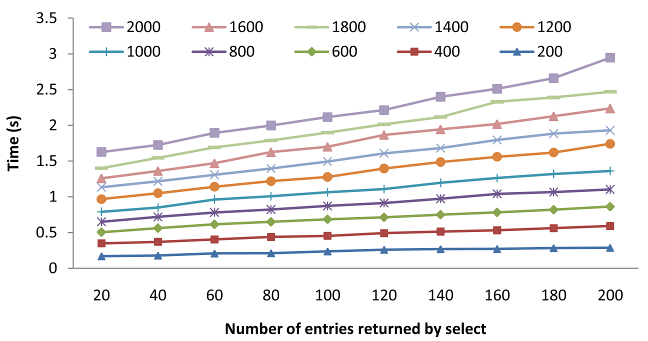

Fig. 10. Time to select a subset of the registered service metadata from MDS with increasing number of total registrations.

5.2 Auction Allocation

The social storage cloud auction mechanism relies on a collection of web services representing the parties involved in the marketplace. A single AM conducts the auction and a single AgM creates SLAs as a result of the auction. Each storage service is represented by a BA which consults local policy to determine a price based on predefined metrics. The major point of stress in this system is the AM and AgM. The AM is responsible for creating an auction, advertising the auction to suitable bidders, soliciting bids, and determining the result of the auction. Agreement creation is simpler as it only involves creation of a WS-Agreement document and one call to the winning bidder to verify the agreement.

Fig. 11 shows the auction throughput with an increasing number of bidders in each auction. The number of auctions per minute is calculated based on the time taken for 500 auctions to complete, this time is measured on the client side starting when the client submits the first task (of 500) through to the creation of the final agreement by the AgM. Fig. 11 represents the worst case situation when all auctions are started immediately; auctions close as soon as all bidders have bid. It is important to remember that in a typical scenario auctions are created with a predefined deadline and users expect some latency between submission and agreement creation. Additionally in a storage context one would expect relatively long term, stable reservations which implies users would not conduct auctions frequently. These results show that even with 50 bidders a small scale AM and AgM can complete 65 auctions per minute which, under our assumptions, would be capable of supporting a large scale social storage cloud. This number could also be increased by adding additional AMs to the system, which would be run independently on dedicated hosts.

5.3 Service Overhead

The final aspect of this evaluation focuses on the overhead of hosting a Social Cloud. As discussed in Section 2.2.6 one

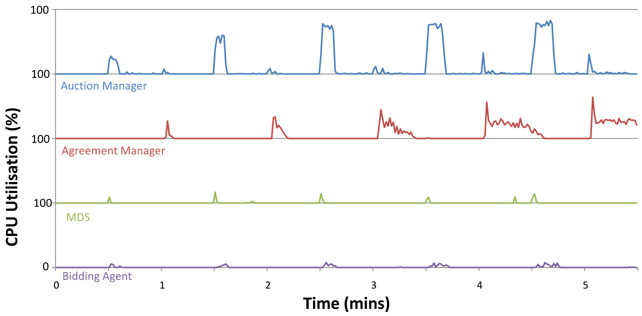

Fig. 12. CPU usage.

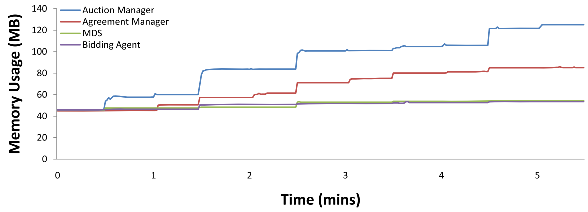

Fig. 13. Memory usage.

way to provide this infrastructure is through the use of a scalable co-op architecture in which members of a Social Cloud contribute services to provide core management functionality. However, to do this it is desirable if the overhead of hosting individual services is not significant. A posted price co-op model would require the distribution of registration and discovery information (MDS), and also agreement management (AgM) functionality across peers of the social network. In a dynamic auction marketplace the auctioneer (AM) also needs to be distributed. In all cases the distributed services must be trusted to maintain data integrity—in the absence of absolute trust secure protocols can be used to establish guarantees over the actions of these services [29].

Each service has been individually monitored to measure the CPU and Memory usage. The following experiments are based on a virtualized testbed deployment, each of the services analyzed is hosted on a dedicated host so as to minimize competition for resources. Twenty bidders have been deployed to simulate the requirements of a realistic auction scenario and the bidders all implement a random bidding policy to distribute allocation. Each service request uses 100 percent of the resources available by a single provider, therefore many of the winning auctions will be unable to be satisfied after the auction completes. In the case a provider is unable to satisfy the requirements of a winning bid the AgM will attempt to iteratively satisfy the requirements by requesting substitute winners [45] from the AM. Auctions have a duration of 30 seconds. In this experiment, the number of simultaneous requests is increased every minute by 10, ranging from 10 requests at time 0:30 through to 50 requests at 4:30.

5.3.1 CPU Usage

Fig. 12 shows the CPU usage of each service for the duration of the sample workload. As the number of requests submitted increases the peak CPU usage of all services and the duration of usage increases. The order of events between services is evident in the graph: as requests are submitted the CPU usage of the AM peaks, which in turn generates work for MDS (to discover bidders) and the Storage Service BA (computing a bid). After the auction period of 30 seconds has elapsed the AM computes a winner and the AgM creates an SLA. As the AgM is operating there is an additional peak in the AM due to result retrieval and the computation of substitute providers. The CPU usage of the AM and AgM use up to 80 percent capacity of the test machine (when 50 simultaneous requests are submitted), however the usage is for a short period of time even under this dense workload. The AM uses the most CPU of the services examined, approaching 100 percent. This is due to the complexity of creating and advertising auctions and also soliciting bids.

Without substitute computation AgM utilization is relatively low and short duration, however as the number of simultaneous requests increases the duration of AgM CPU utilization also increases. This duration is exaggerated due to the computation of substitute providers—all auctions are run concurrently which results in each provider bidding on each auction when in reality they can only host a single task each. MDS usage is minimal for the duration of the workload as it is only used to query for 20 registered bidders. The Bidding Agent computation is shown to be low (below 25 percent). The breadth of usage increases as the number of auctions increases, this is due both to bid computation and agreement confirmation (with increased substitutes).

5.3.2 Memory Usage

The memory usage of each service is shown in Fig. 13. Neither MDS or the BA exceed their initial allocation with maximum memory usage of approximately 50 MB. This is because the BA stores no state directly and MDS only stores metadata for 20 registered bidders. Both the AM and AgM show increased memory usage with the number of jobs submitted. Memory usage peaks at 125 and 85 MB, respectively. The increase in memory usage is due to the amount of auction and agreement state stored by each service and the complexity of the auction process. Due to the limited duration of this experiment memory usage does not decrease as the specified WSRF Resource lifetimes [46] do not expire within this time frame. These results highlight the small footprint of the services even under moderate load.

5.3.3 Summary

The CPU and memory footprint of the core market services was shown to be relatively low and generally short duration for up to 50 simultaneous auctions. The AM exhibited the highest overhead, utilizing almost 100 percent CPU for 10 seconds and 125 MB of RAM with 50 concurrent auctions and 20 bidders, however this scenario represents a worst case scenario when all auctions are started concurrently. For a moderately sized Social Cloud one would expect to host far fewer auctions over a much longer timespan which would therefore utilize less resources for shorter periods of time, in addition the WSRF resource lifetime would ensure memory usage is reduced periodically. The posted priced market was shown to require negligible resources due to the low requirements of MDS with few cloud participants and the low AgM overhead when substitutes are not used.

6 Reflective Analysis

The social storage cloud provides a first step toward realizing the vision of Social Cloud computing. In particular, it provides an integrated platform on which social network friends can trade a single resource (storage) with one another using a credit model. The cloud model was shown to be well suited to this type of scenario as users are able to lease capacity through standardized service-based interfaces using an abstracted virtualized resource layer. This section discusses how the social storage cloud fulfils the high-level vision of Social Cloud computing.

The relationships and policies represented in the social storage cloud are one dimensional, in that all friends are treated equally. At present all users belong to the same group and there is no ability to define different sharing policies based on relationship type. The architecture is able to select users based on their friend relationship and could be extended to retrieve users based on their group membership or relationship type. The evaluation showed the cost of friend selection in MDS to be low for moderately sized Social Clouds. Simple policies are supported in the auction scenario to alter the bid price based on the identity of the requester.

The social storage cloud uses an SLA approach to define requirements and obligations of a trade. In practice the use of SLAs or "contracts" between participants involved in resource sharing within a social context may not be necessary. This is primarily due to the existing level of trust that already exists between participants within a social network. Therefore, an SLA should at most represent a best effort agreement between friends. Due to the nature of sharing, the participants of the social storage cloud do not explicitly consider risk, however a notion of risk can be incorporated into their pricing models if required. However, if a social network involves sharing between participants with varying degrees of trust, an SLA would be a useful capability to support.

The social marketplace currently supports two distinct and independent market protocols to demonstrate simultaneous allocation. While both of these protocols are economically focused the market can be adapted to deploy more novel social protocols.

Supplying infrastructure to a Social Cloud is seen as one of the major hurdles for the creation of a stable Social Cloud due to the reliance on the goodwill of the participants. However, a co-op market model can overcome this limitation due to the minimal overhead of the individual allocation services—even when hosting a complex auction process. Moreover, a co-op does not necessarily rely entirely on social incentives as trustworthy protocols can be used to provide fairness guarantees.

7 Related Work

The term Social Cloud has been used previously to describe different concepts. Originally Google's Kevin Marks defined the Social Cloud as a federated view of social networks [47] provided through OpenSocial [48]. The cloud therefore encapsulates different social networks and provides user transparency between these networks. In work concurrent to ours, Pezzi [49] proposes a Social Cloud as a means of cultivating collective intelligence and facilitating the development of self-organizing, resilient communities. In this vision, the social network and its services are provided by network nodes owned by members of the network rather than by centralized servers owned by the social network. Pezzi's work is in its infancy and has no architectural details or implementation.

Cloud integration with social networks has also been previously explored, however these approaches rely on cloud platforms to host social networks, or create scalable applications within the social network. For example, Facebook users can build scalable cloud-based applications hosted by Amazon web services [50]. ASPEN [51], and PolarGrid [52] also leverage social networking information and applications in their distributed applications.

In scientific domains social networks are increasingly used to coordinate research communities, two such examples are MyExperiment for biologists and nanoHUB.org for the nanoscience community. MyExperiment provides a virtual research environment where collaborators can share research and execute scientific workflows remotely. nanoHUB.org allows users to share data as well as transparently execute applications on distributed resource providers such as TeraGrid. While similar to a Social Cloud, MyExperiment and nanoHUB.org each have specific sharing focuses and build their own proprietary social network.

Volunteer computing is a distributed computing model in which users donate computing resources to a specific (academic) project. The first volunteer project was the Great Internet Mersenne Prime Search (http://www.mersenne.org) in 1996, however the term gained much exposure through the SETI@Home [53], Folding@home [54], and Storage@Home [55] projects in the late 1990s. These projects showed the enormous computing power available through collaborative systems. The focus of Volunteer computing has since shifted toward generic middleware providing a distributed infrastructure independent of the type of computation, for example, BOINC [13]. Most Volunteer platforms do not define any form of SLA, users are typically anonymous and are not accountable for their actions (they are rewarded with different incentives however). In a Social Cloud context this does not suffice as users must have some level of accountability. A more realistic model for this type of open sharing is a credit-based system in which users earn credits by contributing resources and then spend these credits when using other resources. This type of policy is used in systems such as PlanetLab [56].

There is a great deal of synergy between a Social Cloud and a P2P network in that services are provided by a network of peers. There are multiple examples of P2P storage networks in which storage is hosted among a pool of distributed peers [57], [58], [59], P2P networks have also been applied to generic cloud discovery and management [60]. However, there are significant differences between these two paradigms. Unlike a P2P network, Social Clouds exist within the context of a wider social network and are formed based on the encoded relationships. Typically in a P2P storage network users are unable to select storage location or influence the network topology. There is also no notion of social trust, incentive engineering, or market metaphors to facilitate and regulate sharing. In addition management and interaction within Social Clouds follows a group-based cloud model rather than the completely decentralized model used in P2P networks.

FriendStore [61] and AmazingStore [62] are examples of commercial social storage systems. FriendStore is a P2P replication service, in which users backup data on a subset of user selected peers. Rather than leveraging a Social network graph, FriendStore exists outside a particular network, instead it allows users to select trusted nodes from any Social network provider. FriendStore focuses on long term replicated storage and therefore requires complex protocols to store data as separate encrypted chunks. AmazingStore augments managed centralized storage with a P2P network of *Cloudlets* deployed on participating user's machines. Unlike the Social Cloud both FriendStore and AmazingStore offer a single resource (storage), they also create a separate P2P network among members and therefore do not leverage existing Social networks to facilitate sharing. Incentive mechanisms in both cases are limited to reciprocation.

Perhaps the most similar application to the Social Cloud is Intel's "progress thru processors"[5] Facebook application. Using this application users can contribute excess compute power to individually selected scientific projects through Facebook. Users are not rewarded for their contribution as such, however they can view and publish statistics of their contributions. Upon joining the application users may post information to their news feed, or inform friends of the application. The progress thru processors application relies on a generic resource layer constructed by deploying a BOINC application on the users machine.

8 Vision and Future Work

In addition to the definition in Section 1, there are certain characteristics that underpin our vision of what a Social Cloud is and should be. A Social Cloud should need little user expertise to access resources and must therefore exhibit a high degree of access and distribution transparency. A Social Cloud ought to have low barriers for participation—and therefore vastly increase public access to computing, storage, and services. A Social Cloud should allow overlapping groups—with members belonging to multiple groups and thereby (to a limited extent) permit the osmosis of resources across groups based on the social relationships and standing of other members. However, the most critical characteristic is that a Social Cloud uses social relationships to ensure desirable behavior within the system. While these characteristics will be common to many Social Clouds, the characteristics they ultimately embody will depend on the aims and requirements of the individual groups that constitute a Social Cloud.

8.1 Future Work

The Social Cloud presents a rich environment for future research. One major area of future work is adapting the market protocols discussed in Section 2.2.5 to a social context and also looking at other ways to define and exploit social incentives (and disincentives) in a resource sharing scenario. This may involve altering existing protocols or defining new socially oriented trading protocols. Section 3 summarized a range of application scenarios in which a Social Cloud would be appropriate. Of these scenarios, we are currently working on projects to: construct a social computation cloud that permits trading of virtual machine images, build a Social Cloud for public scientific computing (volunteer), and use Social Clouds to support scientific collaboration. We are also exploring the idea of using reputation to measure social compliance in the context of the Social Cloud to ease the "social accounting" that will be incurred as groups grow in size and role. We are also looking further at business models [63] that could be realized in the Social Cloud.

In parallel to these efforts we plan to deploy the social storage cloud to provide a platform for further experimentation. In particular, we aim to explore system performance and user interactions on a much larger scale. This deployment could also be used to examine storage and replication algorithms, and address potential security implications.

9 Conclusion

This paper has presented the vision of Social Cloud computing, an amalgamation of cloud computing and social networking. A Social Cloud is unique in that it builds upon the social incentives and external real-world relationships inherent in social networks to provide heterogeneous resource trading. This work represents a novel approach to collaborative computing utilizing socially corrective mechanisms to motivate contribution and compliance without requiring extensive incentive and enforcement architectures.

A Facebook-based social storage cloud has been developed and deployed. The social storage cloud supports storage trading through a two protocol social marketplace. The integrated social storage Facebook application allows users to discover and trade storage contributed by their friends, taking advantage of preexisting trust relationships. A credit-based trading approach has been adopted to discourage free loading.

It was shown empirically that the marketplaces used for trading and/or reciprocation of services could be hosted using small scale resources, based upon the observation that individual groups are small in size (averaging 130 individuals). In addition, it was shown even under load, the system can perform multiple concurrent auctions that would satisfy the requirements for a moderately sized social network. Finally, the overhead of the Social Cloud services was shown to be small under realistic load conditions, thereby verifying the assertion that a co-op model can be employed to enable a scalable self-contained Social Cloud.

10 Additional Resources

Find the project on Facebook: http://www.facebook.com/SocialCloudComputing (active at the time of publication, December 2012).

5. http://www.facebook.com/progressthruprocessors, last accessed April 2011.

REFERENCES

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. High Performance Computing Applications*, vol. 15, pp. 200-222, Aug. 2001.

[2] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A Privacy-Preserving Online Social Network Leveraging on Real-Life Trust," *IEEE Comm. Magazine*, vol. 47, no. 12, pp. 94-101, Dec. 2009.

[3] O. Nov, M. Naaman, and C. Ye, "Analysis of Participation in an Online Photo-Sharing Community: A Multidimensional Perspective," *J. Am. Soc. for Information Science and Technology*, vol. 61, no. 3, pp. 555-566, 2010.

[4] K.R. Lakhani and E. von Hippel, "How Open Source Software Works: Free User-to-User Assistance," *Research Policy*, vol. 32, pp. 923-943, 2003.

[5] K.R. Lakhani and R.G. Wolf, "Why Hackers Do What They Do," *Perspectives in Free and Open-Source Software*, pp. 3-22, MIT, 2005.

[6] L. Torvalds and D. Diamond, *Just for Fun: The Story of an Accidental Revolutionary*. Harper Business, 2001.

[7] M.M. Wasko and S. Faraj, "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice," *MIS Quarterly*, vol. 29, no. 1, pp. 35-57, 2005.

[8] S.L. Bryant, A. Forte, and A. Bruckman, "Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia," *Proc. Int'l ACM SIGGROUP Conf. Supporting Group Work (GROUP '05)*, pp. 1-10, 2005.

[9] N. Li, *Social Capital: A Theory of Social Structure and Action*. Cambridge Univ., 2002.

[10] N.B. Ellison, C. Steinfield, and C. Lampe, "The Benefits of Facebook Friends: Social Capital and College Students Use of Online Social Network Sites," *J. Computer-Mediated Comm.*, vol. 12, no. 4, pp. 1143-1168, 2007.

[11] P. Resnick, "Beyond Bowling Together: Sociotechnical Capital," *HCI in the New Millennium*, pp. 247-272, Addison-Wesley, 2002.

[12] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: An Experiment in Public-Resource Computing," *Comm. ACM*, vol. 45, no. 11, pp. 56-61, 2002.

[13] D.P. Anderson, "BOINC: A System for Public-Resource Computing and Storage," *Proc. IEEE/ACM Fifth Int'l Workshop Grid Computing (GRID '04)*, pp. 4-10, 2004.

[14] M. Harvey, G. Giupponi, J. Villà-Freixa, and G. De Fabritiis, "PS3GRID.NET: Building a Distributed Supercomputer Using the PlayStation 3," *Distributed and Grid Computing - Science Made Transparent for Everyone. Principles, Applications and Supporting Communities*, Rechenkraft.net, 2007.

[15] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, and F. Players, "Predicting Protein Structures with a Multiplayer Online Game," *Nature*, vol. 466, no. 7307, pp. 756-760, 2010.

[16] D. Stainforth, J. Kettleborough, A. Martin, A. Simpson, R. Gillis, A. Akkas, R. Gault, M. Collins, D. Gavaghan, and M. Allen, "Climateprediction.net: Design Principles for Public-Resource Modeling Research," *Proc. 14th Int'l Assoc. Science and Technology for Development (IASTED) Int'l Conf. Parallel and Distributed Computing and Systems*, pp. 32-38, 2002.

[17] N. Andrade, F. Brasileiro, M. Mowbray, and W. Cirne, "A Reciprocation-Based Economy for Multiple Services in a Computational Grid," *Market Oriented Grid and Utility Computing*, R. Buyya and K. Bubendorfer, eds., pp. 357-370, Wiley, 2009.

[18] N. Andrade, W. Cirne, F. Brasileiro, and P. Roisenberg, "OurGrid: An Approach to Easily Assemble Grids with Equitable Resource Sharing," *Proc. Ninth Workshop Job Scheduling Strategies for Parallel Processing*, June 2003.

[19] H. Park and M. van der Schaar, "Evolution of Resource Reciprocation Strategies in P2P Networks," *IEEE Trans. Signal Processing*, vol. 58, no. 3, pp. 1205-1218, Mar. 2010.

[20] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation Systems," *Comm. ACM*, vol. 43, pp. 45-48, Dec. 2000.

[21] A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, pp. 618-644, Mar. 2007.

[22] J. Sabater and C. Sierra, "Reputation and Social Network Analysis in Multi-Agent Systems," *Proc. First Int'l Joint Conf. Autonomous Agents and Multiagent Systems: Part 1 (AAMAS '02)*, pp. 475-482, 2002.

[23] P. Wu and G. Wu, "Reputation Mechanism in Peer-to-Peer Network," *Proc. First Int'l Conf. Information Science and Eng. (ICISE)*, pp. 1793-1796, Dec. 2009.

[24] L. Mingbiao, L. Jian, and X. Shengli, "Posted Price Model Based on GRS and Its Optimization Using in Grid Resource Allocation," *Proc. Int'l Conf. Wireless Comm., Networking and Mobile Computing*, pp. 3172-3175, 2007.

[25] S. Seifert, *Posted Price Offers in Internet Auction Markets*. Springer, 2006.

[26] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta, "Spawn: A Distributed Computational Economy," *IEEE Trans. Software Eng.*, vol. 18, no. 2, pp. 103-117, Feb. 1992.

[27] R. Buyya and S. Venugopal, "The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report," *Proc. IEEE First Int'l Workshop Grid Economics and Business Models (GECON '04)*, 2004.

[28] W.-Y. Lin, G.-Y. Lin, and H.-Y. Wei, "Dynamic Auction Mechanism for Cloud Resource Allocation," *Proc. IEEE/ACM 10th Int'l Conf. Cluster, Cloud and Grid Computing (CCGrid)*, pp. 591-592, 2010.

[29] K. Chard and K. Bubendorfer, "Using Secure Auctions to Build a Distributed Meta-Scheduler for the Grid," *Market Oriented Grid and Utility Computing*, Wiley Series on Parallel and Distributed Computing, R. Buyya and K. Bubendorfer, eds., pp. 569-588, Wiley, 2009.

[30] D. Abramson, J. Giddy, and L. Kotler, "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?" *Proc. 14th Int'l Symp. Parallel and Distributed Processing (IPDPS '00)*, pp. 520-528, 2000.

[31] M. Mattess, C. Vecchiola, and R. Buyya, "Managing Peak Loads by Leasing Cloud Infrastructure Services from a Spot Market," *Proc. IEEE 12th Int'l Conf. High Performance Computing and Comm. (HPCC)*, pp. 180-188, 2010.

[32] K. Bubendorfer, I. Welch, and B. Chard, "Trustworthy Auctions for Grid-Style Economies," *Proc. IEEE Sixth Int'l Symp. Cluster Computing and the Grid (CCGRID '06)*, pp. 386-390, 2006.

[33] K. Suzuki and M. Yokoo, "Secure Generalized Vickery Auction Using Homomorphic Encryption," *Proc. Seventh Int'l Conf. Financial Cryptography (FC '03)*, pp. 239-249, 2003.

[34] M. Naor, B. Pinkas, and R. Sumner, "Privacy Preserving Auctions and Mechanism Design," *Proc. First ACM Conf. Electronic Commerce (EC '99)*, pp. 129-139, 1999.

[35] K. Bubendorfer, B. Palmer, and W. Thomson, "Trust in Grid Resource Auctions," *Market Oriented Grid and Utility Computing*, Wiley Series on Parallel and Distributed Computing, R. Buyya and K. Bubendorfer, eds., pp. 541-568, Wiley, 2009.

[36] D.D. Roure, C. Goble, and R. Stevens, "The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows," *Future Generation Computer Systems*, vol. 25, no. 5, pp. 561-567, 2009.

[37] G. Klimeck, M. McLennan, S.P. Brophy, G.B. Adams III, and M.S. Lundstrom, "nanoHUB.org: Advancing Education and Research in Nanotechnology," *Computing in Science and Eng.*, vol. 10, pp. 17-23, Sept. 2008.

[38] E. Hammer-Lahav, D. Recordon, and D. Hardt, "OAuth 2.0 Authorization Protocol," IETF Internet draft, 2011.

[39] A. Andrieux et al., *Web Services Agreement Specification (WS-Agreement)*, 2007.

[40] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid Information Services for Distributed Resource Sharing," *Proc. 10th IEEE Symp. High Performance Distributed Computing (HPDC)*, 2001.

[41] K. Czajkowski, D.F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe, "The WS-Resource Framework," Globus, technical report, http://www.globus.org/wsrf/specs/ws-wsrf.pdf, 2004.

[42] D. Neumann, J. Stößer, A. Anandasivam, and N. Borissov, "SORMA - Building An Open Grid Market for Grid Resource Allocation," *Proc. The Fourth Int'l Workshop Grid Economics and Business Models (GECON '07)*, pp. 194-200, 2007.

[43] N. Borissov, S. Caton, O. Rana, and A. Levine, "Message Protocols for Provisioning and Usage of Computing Services," *Proc. Sixth Int'l Workshop Grid Economics and Business Models*, pp. 160-170, 2009.

[44] A. Anjomshoaa et al., "Job Submission Description Language (JSDL) Specification, Version 1.0," 2005.

[45] K. Chard, K. Bubendorfer, and P. Komisarczuk, "High Occupancy Resource Allocation for Grid and Cloud Systems, a Study with Drive," *Proc. 19th ACM Int'l Symp. High Performance Distributed Computing (HPDC '10)*, pp. 73-84, 2010.

[46] L. Srinivasan and T. Banks, "Web Services Resource Lifetime 1.2," OASIS, OASIS Standard, http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf, 2006.

[47] K. Marks, "Future of Web APIs: The Social Cloud," *Future of Web Apps,* 2008.

[48] OpenSocial and Gadgets Specification Group, "OpenSocial Specification V0.9," http://www.opensocial.org/Technical-Resources/opensocial-spec-v09/OpenSocial-Specification.html, Apr. 2009.

[49] R. Pezzi, "Information Technology Tools for a Transition Economy," http://www.socialcloud.net/papers/ITtools.pdf, Sept. 2009.

[50] Amazon, "Develop and Scale Facebook Apps with AWS," http://aws.amazon.com/solutions/global-solution-providers/facebook, 2012.

[51] R. Curry, C. Kiddle, N. Markatchev, R. Simmonds, T. Tan, M. Arlitt, and B. Walker, "Facebook Meets the Virtualized Enterprise," *Proc. IEEE 12th Int'l Enterprise Distributed Object Computing Conf. (EDOC '08),* pp. 286-292, 2008.

[52] Z. Guo, R. Singh, and M. Pierce, "Building the PolarGrid Portal Using Web 2.0 and OpenSocial," *Proc. Fifth Grid Computing Environments Workshop (GCE '09),* pp. 1-8, 2009.

[53] D. Werthimer, J. Cobb, M. Lebofsky, D. Anderson, and E. Korpela, "SETI@HOME—Massively Distributed Computing for SETI," *Computing in Science and Eng.,* vol. 3, no. 1, pp. 78-83, 2001.

[54] M.R. Shirts and V.S. Pande, "Screensavers of the World Unite!" *Science,* vol. 290, pp. 1903-1904, 2000.

[55] A.L. Beberg and V.S. Pande, "Storage@home: Petascale Distributed Storage," *Proc. IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS),* pp. 1-6, 2007.

[56] L. Peterson and T. Roscoe, "The Design Principles of PlanetLab," *SIGOPS Operating Systems Rev.,* vol. 40, no. 1, pp. 11-16, 2006.

[57] A. Rowstron and P. Druschel, "Storage Management and Caching in Past, a Large-Scale, Persistent Peer-to-Peer Storage Utility," *SIGOPS Operating Systems Rev.,* vol. 35, pp. 188-201, Oct. 2001.

[58] L.P. Cox and B.D. Noble, "Samsara: Honor among Thieves in Peer-to-Peer Storage," *SIGOPS Operating Systems Rev.,* vol. 37, pp. 120-132, Oct. 2003.

[59] A. Lakshman and P. Malik, "Cassandra: Structured Storage System on a P2P Network," *Proc. 28th ACM Symp. Principles of Distributed Computing (PODC '09),* p. 5, 2009.

[60] R. Ranjan, L. Zhao, X. Wu, A. Liu, A. Quiroz, and M. Parashar, "Peer-to-Peer Cloud Provisioning: Service Discovery and Load-Balancing," *Cloud Computing,* Computer Communications and Networks, N. Antonopoulos and L. Gillam, eds., pp. 195-217, Springer, 2010.

[61] D.N. Tran, F. Chiang, and J. Li, "Friendstore: Cooperative Online Backup Using Trusted Nodes," *Proc. First Int'l Workshop Social Network Systems (SocialNet '08),* 2008.

[62] Z. Yang, B.Y. Zhaoy, Y. Xing, S. Ding, F. Xiao, and Y. Dai, "AmazingStore: Available, Low-Cost Online Storage Service Using Cloudlets," *Proc. Ninth Int'l Conf. Peer-to-Peer Systems (IPTPS '10),* p. 2, 2010.

[63] O. Rana and S. Caton, "Business Models for On-Line Social Networks: Challenges and Opportunities," *Int'l J. Virtual Communities and Social Networking (IJVCSN),* vol. 2, no. 3, 2010.

Kyle Chard received the BSc (Hons.) degree in computer science, the BSc degree in mathematics and electronics, and the PhD degree in computer science from the Victoria University of Wellington in 2011. He is a senior researcher at the Computation Institute, University of Chicago, and Argonne National Laboratory. His research interests include distributed meta-scheduling, grid and cloud computing, economic resource allocation, social computing, services computing, and medical natural language processing. He is a member of the IEEE.

Kris Bubendorfer received the PhD degree on mobile agent middleware in computer science from the Victoria University of Wellington in 2002. He is the program director for networking engineering and a senior lecturer in the School of Engineering and Computer Science at the Victoria University of Wellington. His research interests include market-oriented utility computing, social computing, digital provenance, and reputation. He is a member of the IEEE.

Simon Caton received the PhD degree with a focus on autonomic management of volunteer resources for distributed and parallel image processing in campus-grids from Cardiff University in 2010. He is a senior researcher at the Karlsruhe Service Research Institute and the head of the Corporate and Cloud Services Research Division at the Institute for Information Systems and Management, both at Karlsruhe Institute of Technology. Prior to that, he was a research associate in the School of Computer Science and Informatics, Cardiff University, where he worked on the FP6 EU Project SORMA in the domain of SLA management and enforcement. His research interests include Social Cloud computing, the adoption of autonomic computing for electronic markets, service value networks, and service level agreements. He is a member of the IEEE.

Omer F. Rana received the PhD degree in neural computing and parallel architectures from the Imperial College of Science, Technology, & Medicine, London University, United Kingdom. He is a professor of performance engineering in the School of Computer Science & Informatics, Cardiff University, United Kingdom. His research interests include high performance distributed computing, data mining, and multiagent systems. He served as the deputy director of the Welsh e-Science Centre and has also worked as a software developer in the United Kingdom. He is a member of the IEEE.