Social Network Storage Allocation

Faruk Bagci
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
dr.faruk.bagci@gmail.com

Mohamed Esam
Department of Information Engineering & Technology
German University in Cairo
Cairo, Egypt
mohamed.esameldin@guc.edu.eg

Aya Kamel, Iman Mansour, Rimon Hanna, Seif El-Din Allam, Taher Galal
Department of Media Engineering and Technology
German University in Cairo
Cairo, Egypt
{aya.kamel, iman.mansour, rimon.hanna, seif.allam, taher.galal}@student.guc.edu.eg

Abstract

Today, social networks present massive amounts of data by the hour that need storage, therefore, along with the aid of cloud computing, social networks users can have their data stored in data centers anywhere around the globe belonging to the cloud. This paper will be focusing on how to allocate user data to the appropriate global data centers from a social networking point of view. The method is carried out using the proposed algorithm where a number of factors are involved such as; read-rate, write-rate, and the number/location of friend connections are used to calculate which data center would yield shorter latency and therefore better results if the user data was to be stored at that location. After validating which was done via simulation, the algorithm proved to yield sufficiently improved data-access latency scores in all test cases.

Keywords: social networks, storage allocation, cloud computing

1. Introduction

In the present day, the use of social networks as means of communication have almost become a part of people's everyday routine for generations both young and old. The online social network industry has boomed over the last decade due to the high demand of the willing public. The modern internet-using public currently depends on social networks to maintain relations with friends, family, work related contacts, and even business marketing while also keeping track of events, meetings, etc. Social networks could be divided into either purely social (e.g. Facebook) or business driven (e.g. LinkedIn) each serving different purposes. Social networks are made to support massive amounts of data traffic, in which users post, upload photos or videos, and connect with friends. Furthermore, many businesses nowadays are turning to social networks, such as Facebook or Twitter, as major marketing tools, where they can create 'fan' pages and monitor various aspects regarding the business such as; keeping track of positive or negative comments regarding the company or its products, and spotting new product/service opportunities [1] [2]. Considering all the services available in social networks from uploading pictures on Flickr, videos on Youtube, joining a project via LinkedIn or simply posting to your friends on Facebook, all this data needs to be stored 'somewhere'. That 'somewhere' is the cloud [3].

Cloud computing as a system is a complex combination of hardware, software, storage and processing distributed around the globe which work together to form one major entity, i.e. the cloud. The system allows a user connected to the cloud to immediately access latter resources wherever their location is [3]. Therefore, instead of using stand-alone servers, each with its own individual resources and storage, the cloud offers multiple units which hold thousands of computers, storage devices, and networks performing. These units are known as data centers. The process of making these data centers, and the vast numbers of machines of which they are comprised of, able to be viewed as one single entity (cloud) to the user is done by virtualization. With virtualization as a tool, the cloud can provide users with virtual storage to store user information, create virtual networks to connect clients, as well as virtual servers to process the vast amounts of data traffic being transmitted through the cloud. As a result, the cloud offers storage, processing and many other features to various users without burdening the user on where and how her data is to be handled and stored.

Finally, putting in mind that clouds are currently the main storage system for most social networks, where all users are instantly able to share the pool of resources which the cloud offers. This combination gave rise to the question of which data (belonging to which user) should be stored in which location (data center).

The aim of this paper is to improve data retrieval latency i.e. the time taken to retrieve the data required from the data center to the demanding user and vice versa and therefore improve overall performance of the storage system in the social networking context.

2. Related Work

As the social networking systems are growing more and more popular, it becomes more and more important to start looking at how the data is being stored [4]. Moreover as social networking started to give extra features such as games, video sharing and storing all of the users' statuses, the need increases to have better means of storing such enormous amounts of data.

New methods of data storage needed to be implemented. Here arises the use of cloud computing [5], allowing the ability to have more data centers located over different parts of the world, while having all these data centers connected together, and giving rise to easier means of communication between the data centers. This is mainly because different people's data will be stored over different data centers, but it could be for some reason that someone needs access to data of her friend which lives in a different country and has her data stored on a different data center [6].

Here the cloud computing technology comes in handy. As we can have all data centers connected together in one large cloud therefore wherever the data of the user lies it can be accessed by herself or any of her friends. Also the use of social networking increases the efficiency of data flow. Therefore the use of new services should arise so that could add extra features to the user.

Some research was first performed on the topic of social networking and cloud computing. In [7] the main research concentrates on how security issues can arise and how to solve them when information is placed over the cloud. Methods of how to overcome these issues are taken into consideration and tested.

Also, in [8] authors regard the same aspects as before but from a different perspective where practices were analyzed that can be used to maintain a secure system while using social network platforms. Moreover, [9] elaborates how the use of cloud computing would address the utilization of memory and data flow from the perspective of the network communication.

Other research projects elaborate how to increase the usability of social networks and how the use of these social networks would use cloud computing technologies to better share resources between the social network users [10] [11].

In [12] authors specify which place a developer should host her application regarding the geographical position of the application users, and how the cloud technology should handle peeks in the usage of this application.

3. Problem Formulation

The problem this paper aims to solve is basically that the optimum location i.e. data center, to store the user's data is not necessarily the actual location of the user. However, this depends on the location of the user's friends with respect to the user's current location i.e. the users who send or receive data to/from the targeted user.

Figure 1: A case where the optimum location to store user's data is not her actual location

	User 1	User 2	User 3	User 4	User 5
User 1	0	1	0	1	0
User 2	1	0	1	1	0
User 3	0	0	0	1	1
User 4	1	1	1	0	1
User 5	0	0	1	1	0

Table I: Friendship Matrix showing friendship relations between 5 users

Figure 1 shows an example of the latter case where the optimum location to store the user's data is not the actual user location. This example shows a user whose data is stored in her actual location while on the other hand most of this user's friends are logged in from a different location. In this case, every time this user communicates with one of her friends, the data has to propagate all the way from the user's actual location to the friend's location which implies a very high delay and cost. However, if the user's data is stored in the same location of the friend, the data will only propagate within the same data center which is definitely lower in cost.

Table I shown above represents the friendship relationship *f* between users of the social network where $f(a, b) = 1$ when users a and b are friends and n

represents the number of users in the system. Therefore, for a user u_i, the number of friends can be calculated as

$$k = \sum_{j=1}^{n} f(i,j)$$

The measure from which the optimum position will be calculated is the time taken for data transmission between users, called *cost*. Calculating cost depends on various attributes including user's post rate, read rate, and location of both the user and her corresponding friends.

Assuming that *T(a, b)* is the cost of data transfer between data centers *a* and *b*, *l* is the location of u_i, d_i is the data center where user *i*'s data is stored, *post(x)* is the post rate of user *x* and *read(x)* is the read rate of user *x*, then the cost of user u_i in the system can be represented as

$$Cost(u_i) = T(l, d_i) \times post(u_i) + \sum_{j=1}^{k} T(l, d_j) \times read(u_i)$$

The aim of the research presented in this paper is to mitigate the total cost of the system which can be represented by

$$Cost_{total} = \sum_{j=1}^{n} Cost(u_i)$$

constrained by the facts that per data center:

$$\sum_{i=1}^{num} Capacity(u_i) \leq Capacity(datacenter)$$

$$\&\&$$

$$\sum_{i=1}^{num} CPU(u_i) \leq CPU(datacenter)$$

$$\&\&$$

$$\sum_{i=1}^{num} RAM(u_i) \leq RAM(datacenter)$$

where *num* is the number of users whose data is stored in the data center, *Capacity(x)* is amount of storage needed for user x or available in data center *x* (in TB). *CPU(x)* is the processing capabilities/requirements of data center/user *x* and *RAM(x)* is the memory capabilities/requirements of data center/user *x*. The algorithm explained in the following section presents a solution for the latter problem.

4. Algorithm

We are living in the world of Web 2.0, where hundreds of millions of people are connected to the Internet and millions of those people are connected on social networking sites. Facebook for example has a widespread all over the world as shown in Figure 2 which shows the distribution of Facebook users all over the world in white color. Other means of social networks like LinkedIn, MySpace, and Twitter are also massively used as well as blogs, YouTube and Flickr. The vast amount of ways in which people can be connected online has sparked the interest of cloud computing services. Cloud computing services have been developing ways to tap into the Web 2.0 world and establish means of turning the flow of information and communication into business potential.

Figure 2: Distribution of Facebook Users around the World [13]

In order to optimize locating the users' data so as to reduce the cost of retrieving it, the proposed algorithm chooses the optimal data center to store the users' data according to some parameters. According to these parameters, the cost of storing the users' data in each possible data center is calculated and the optimal location is chosen accordingly i.e. the data center with minimum cost.

There are multiple factors affecting this decision. The first factor being the percentages of the user's friends in each country, these percentages is then multiplied by the weight of their distance from the user. The second of these factors is how often the user interacts with her friends and what is the form of these interactions, does she read posts by her friends more or does she create her own posts?

Each of these interactions is assigned a different weight; posting is assigned a bigger weight than reading since the write penalty to a storage device is greater making it essential to reduce this penalty by storing the user's data closer to her location. Last but not least we had to put in consideration the performance differences between the servers and data centers in each country in terms of CPU, RAM, storage capacity, and storage

performance; assigning better weights to servers with higher performance.

Finally all of these factors were combined together to form a performance evaluation algorithm of each of the cases of the user's data being stored in one of the data centers and the data is then moved to the data center with the best performance value. Figure 3 shows the sequence of steps carried out in the algorithm in order to reach the decision which location is the optimum location for storing user's data.

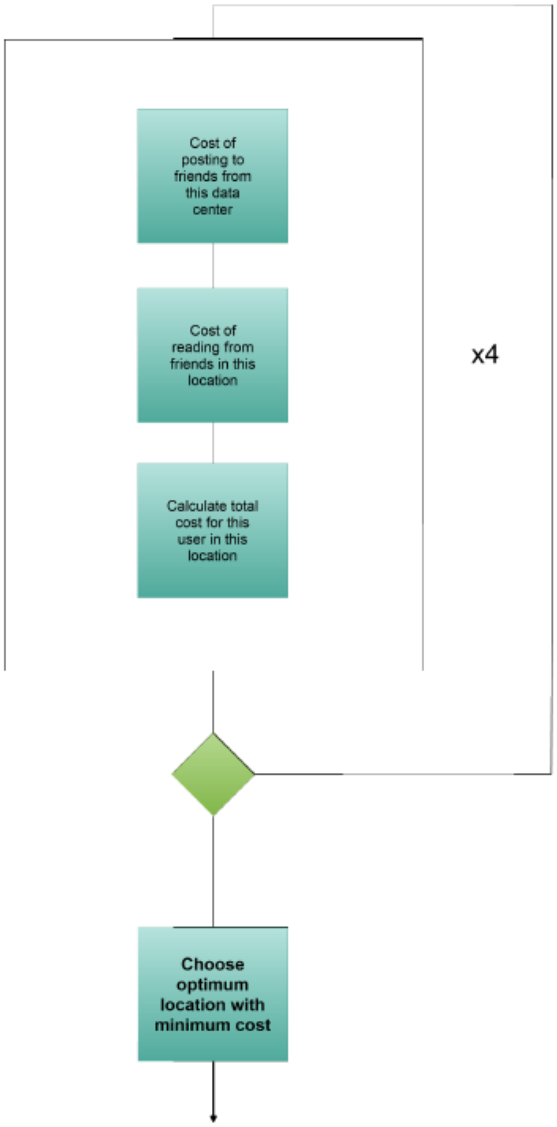

Figure 3: A flow chart showing the proposed algorithm steps per user

The result of applying the algorithm on the environment shown in Figure 1 will be moving the users to the data centers in which most of their friends exist as shown in Figure 4.

5. Implementation

The discussed algorithm needs to be tested on a social network environment where users have friends and can write data and read data of friends. A JAVA model was built in order to simulate the environment on which the algorithm will be applied. In this model, three main objects were created: the user object, the server object, and the data center object.

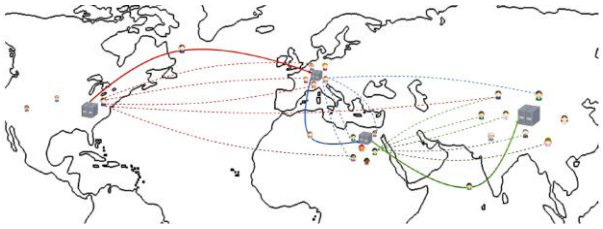

Figure 4: Moving the user data to the optimum location according to the user's friends (moves shown by arrows from the initial location to the destination)

The aim of the user object was to simulate the presence of a real user in the simulation. Therefore, the following properties were created for being able to describe a user: a unique username, the location of the stored user's data, the current location of the user, the rate at which the user reads data of her friends, the rate at which the user writes new data, the required RAM size, the required storage size, and the CPU capability.

As discussed earlier, there was an urge to simulate servers in the environment since they are the ones responsible for processing and overloading the servers was not a valid option. For this purpose, a server object was created. It was simply represented by only two main parameters: The RAM and the CPU where the RAM describes the RAM size of the server and the CPU describes the CPU capability of the server.

The last and one of the most important objects in this environment was the data center. The object was created with the four main properties which describe a data center. These properties are the location of the data center, a list of the servers connected to that data center, a list of the storage arrays of that data center, and a list of users having their data stored in that data center.

After simulating the standalone nodes in the environment, the simulation of the overall system environment was created by creating different instances of these nodes and creating links between them. This was done by generating four instances of the data center object. Each data center was located in a different country: the first one is located in Egypt, the second in USA, the third in Germany and the fourth in China. For each data center, two server objects were attached.

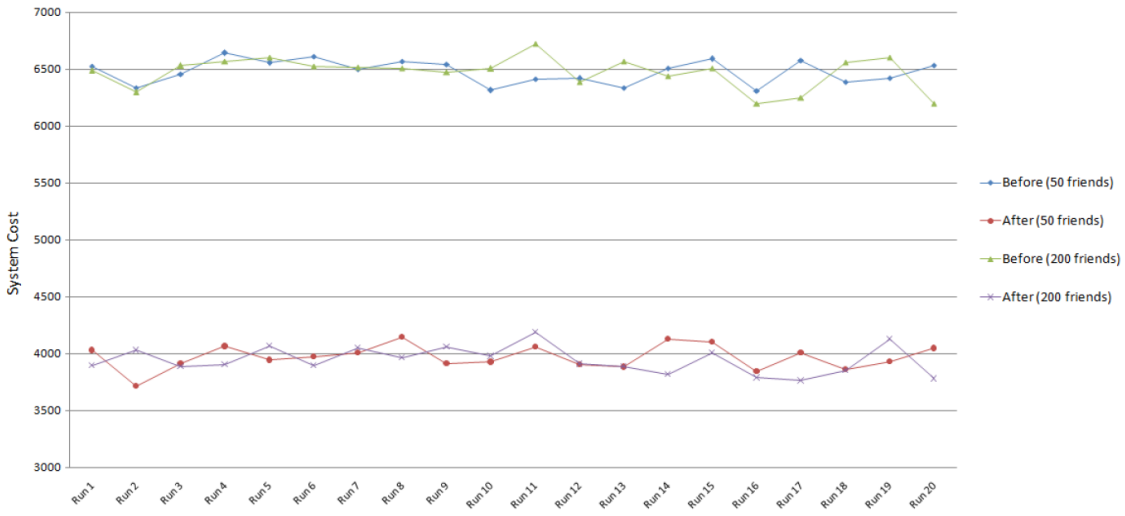

Figure 5: The cost values before and after running the algorithm with 500 users

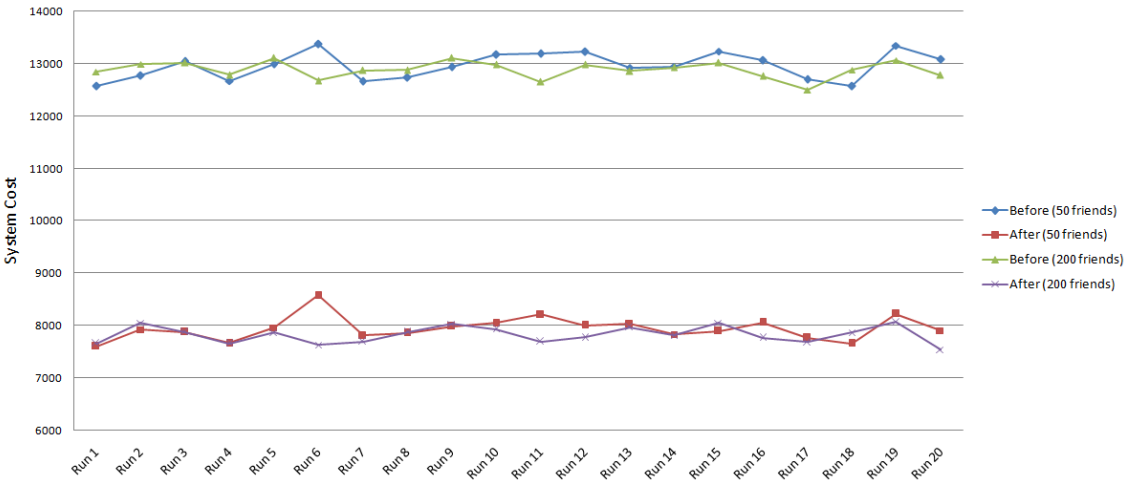

Figure 6: The cost values before and after running the algorithm with 1000 users

Table II shown below presents an overview of the simulation settings that were used when setting up the environment. Users were created randomly using the uniform distribution function *math.rand()* and were also randomly placed in different locations such that initially their data is stored in the nearest data center. Finally, to simulate the idea of a user having friends, a friendship matrix is created indicating which user is friend with which other users.

The last step here was to run the simulation and start collecting results. The results were collected by calculating the cost efficiency using the equations presented in the Problem Formulation section before and after applying the algorithm. To estimate the real impact of the algorithm, the environment is simulated more than once and each time the conditions of the environment run were changed for example the number of users on the system and the number of friends for each user. For each simulation, the process of collecting results was done. The results obtained from the different simulations are presented in the next section.

	Data center				User
	EG	CH	US	DE	
Storage	10TB	15TB	20TB	10TB	10-50 MB
RAM	2x8GB	2x8GB	2x8GB	2x8GB	10MB
CPU	2x22 GHz	2x22 GHz	2x22 GHz	2x22 GHz	10-50 MHz

Table II: The simulation settings

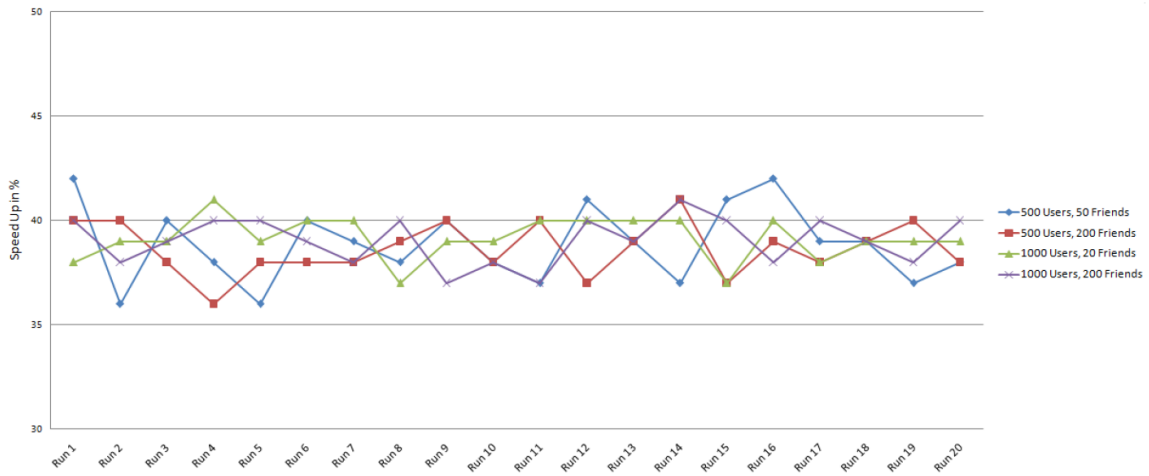

Figure 7: The speed up values calculated from the 4 different system simulations

5. Experimental Results

After the simulation model was complete, some tests were carried out to evaluate the performance of the algorithm. These tests were carried out by calculating the total usage cost for all of the users in the system as shown in the problem formulation section twice; once before running the algorithm when users were assigned to random data centers according to their initial location, and once after running the algorithm when the users data location changed according to the friends location. Dividing the first number by the latter gives the speed up that resulted from the algorithm. These tests were repeated for different number of users with different numbers of friends.

Looking deeper into the simulation results of running the system with 500 users where each user has 50 friends, these simulations yielded speed up results with interesting speed up values ranging from 36% to 42% with an average of 39% and a standard deviation of 1.8. Repeating the same test but with 200 friends per user, the speed up results were also around the same values. Figure 5 shows the cost values before and after running the algorithm.

Increasing the number of users in the system to 1000 users where each user has 50 friends yielded speed up results ranging from 37% to 41% with an average of 39.15% and standard deviation of 1.03. Increasing the number of friends per user to 200 friends, the speed up results were also almost the same. Figure 6 shows the cost values before and after running the algorithm using different numbers of friends per user.

It is of course very logical that the values of cost in the simulation with 500 users as shown in Figure 5 is significantly less than those shown in Figure 6 with 1000 users. The reason is that as mentioned previously, the total cost of the system is the sum of costs of each user in the system. This implies that the cost of the system is directly proportional to the number of users in the simulation.

Having a broad look on the speed up values in Figure 7, it can be noticed that the different runs under several different conditions resulted in speed up values with high precision within the range of 36-42%. This gives an indication that the algorithm is not affected by increasing numbers of users and that applying this algorithm on the huge numbers of social networks users will give more or less 35-40% improvement in the overall system performance.

Moreover, Figure 7 shows that although it was expected that the increasing number of friends will increase the probability of not having a lot of friends in the same location, the algorithm showed a constant response even with the presence of complex friendship matrix between the users.

7. Conclusion

In this paper, the target was to present an algorithm which optimizes the choice of data location for a certain user according to the location of her friends in the social network. To test the efficiency and speed up of the suggested algorithm, a social network environment was simulated and the performance before and after applying the algorithm was calculated.

The results presented in the previous section led to some interesting conclusions. The first and most general conclusion was that locating the friends' data according to her friends' location can be a good approach to optimize the performance of the whole social networking system. This was proved by the

approximately 40% speed up that was achieved after applying the algorithm.

Another conclusion that is specific to the algorithm proposed in this paper is that the performance gain resulting from applying the algorithm is significantly constant. This was proved by the low values of deviation which indicate a high level of precision in the resulting speed ups throughout the different runs and under different simulation conditions e.g. different numbers of users and different numbers of friends per user.

Therefore, it can be concluded that applying the presented algorithm may result in a significant increase in performance which in turn leads to huge cost and power savings as well as a more convenient level of service for the social network users.

8. Limitations and Future Work

The research presented in this paper had some limitations which can be targeted for future research. For example, the maximum number of users used for simulation in this research was 1000 users. Therefore, more tests targeting a larger number of social networks users which in reality exceed millions can be carried out.

Another limitation was that the simulations were carried out on only four data centers with two servers. These numbers do not represent real simulation parameters since in reality the number of servers per data center are definitely more than two servers and there may exists more than four data centers. Therefore, a possible future research is to use an accurate number of data centers and servers as well as using real specification description, e.g. RAM and CPU frequency.

In this research it was assumed that users belong to the countries in which the data centers exist. This assumption led to some location constraints thus some common cases were not thoroughly tested in the performed simulation. Therefore, a target for future research may be extending the simulation model to research more cases like allowing users from all countries not just the ones in which a data center exists.

9. References

[1] Soumitra Dutta Matthew Fraser. The business advantages of social networking. Finance and Management, (168), July/August 2009.

[2] Jeff Bullas. 12 major business benefits of the social media revolution, 2012. http://www.jeffbullas.com/2011/02/14/12-major-benefits-of-the-social-media-revolution/

[3] Rich Maggiani. Cloud computing intersects with social media. The promise of cloud computing, especially as it relates to social media, is considerable, 2012. http://www.solari.net/documents/position-papers/Solari-Cloud-Computing-Intersects-Social-Media.pdf.

[4] Zhong Chen Suke Li. Social services computing: Concepts, research challenges, and directions. IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom) Green Computing and Communications (GreenCom), 2010, pages 840–845, December 2010.

[5] Jamie Syke Bianco. Social networking and cloud computing: Precarious affordances for the "prosumer". WSQ: Women's Studies Quarterly, 37(1,2):303–309, 2009.

[6] EMC Education Services G. Somasundaram, Alok Shrivastava, editor. Information Storage and Management. Storing, Managing, and Protecting Digital Information. Wiley Publishcing, Inc., 2009.

[7] N. Markatchev R. Simmonds Tan Tingxi M. Arlitt B. Walker R. Curry, C. Kiddle. Facebook meets the virtualized enterprise. EDOC '08. 12th International IEEE Enterprise Distributed Object Computing Conference, 2008, pages 286– 292, September 2008.

[8] Ashish S. Prasad. Cloud computing and social media: Electronic discovery considerations and best practices. The Metropolitan Corporate Counsel, pages 26–27, February 2012.

[9] M. Sato. Creating next generation cloud computing based network services and the contributions of social cloud operation support system (oss) to society. WETICE '09. 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009, July 2009.

[10] O. Rana K. Bubendorfer K. Chard, S. Caton. Social cloud: Cloud computing in social networks. IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pages 99–106, July 2010.

[11] K. Chard A. M. Thaufeeg, K. Bubendorfer. Collaborative eresearch in a social cloud. IEEE 7th International Conference on E-Science (e- Science), 2011, pages 224–231, 2011.

[12] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Ching-Hsien Hsu, Laurence Yang, Jong Park, and Sang-Soo Yeo, editors, Algorithms and Architectures for Parallel Processing, volume 6081 of Lecture Notes in Computer Science, pages 13–31. Springer Berlin / Heidelberg, 2010.

[13] Paul Butler. Visualizing friendships, December 2010. http://www.facebook.com/note.php?note id=469716398919.