

Privacy, Cost, and Availability Tradeoffs in Decentralized OSNs

Amre Shakimov
Duke University
Durham, NC
shan@cs.duke.edu

Landon P. Cox
Duke University
Durham, NC
lpcox@cs.duke.edu

Alexander Varshavsky
AT&T Labs
Florham Park, NJ
varshavsky@research.att.com

Ramón Cáceres
AT&T Labs
Florham Park, NJ
ramon@research.att.com

ABSTRACT

Online Social Networks (OSNs) have become enormously popular. However, two aspects of many current OSNs have important implications with regards to privacy: their centralized nature and their acquisition of rights to users' data. Recent work has proposed decentralized OSNs as more privacy-preserving alternatives to the prevailing OSN model. We present three schemes for decentralized OSNs. In all three, each user stores his own personal data in his own machine, which we term a Virtual Individual Server (VIS). VISs self-organize into peer-to-peer overlay networks, one overlay per social group with which the VIS owner wishes to share information. The schemes differ in where VISs and data reside: (a) on a virtualized utility computing infrastructure in the cloud, (b) on desktop machines augmented with socially-informed data replication, and (c) on desktop machines during normal operation, with failover to a standby virtual machine in the cloud when the primary VIS becomes unavailable. We focus on tradeoffs between these schemes in the areas of privacy, cost, and availability.

Categories and Subject Descriptors

C.0 [General]: System Architectures; D.4.6 [Operating Systems]: Security and Protection—*Access Controls*

General Terms

Design, Performance, Reliability, Security

Keywords

Cloud computing, online social networks, privacy, replication, utility computing, virtual machines

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WOSN'09, August 17, 2009, Barcelona, Spain.

Copyright 2009 ACM 978-1-60558-445-4/09/08 ...\$5.00.

1. INTRODUCTION

The popularity of Online Social Networks (OSNs) has experienced unprecedented growth. For example, Facebook [9] attracted over 130 million unique visitors in June 2008, compared to 52 million a year earlier [7]. It had more than 200 million users as of May 2009. Odnoklassniki [14] and V Kontakte [20] have acquired more than 60 million users from the Russian-speaking sector of the Internet alone.

As a result of this popularity, OSN services have amassed large amounts of information about their users, including personal profiles, friend relationships, daily activities, and photos. For instance, Facebook has become the largest photo-sharing service on the Internet, outgrowing even dedicated photo services like Flickr and Photobucket [7].

Unfortunately, two characteristics of current OSNs raise important privacy concerns. One, most OSNs concentrate the data of all their users under a single administrative domain. This concentration makes them vulnerable to large-scale privacy breaches from intentional and unintentional data disclosures. For example, sensitive user data may be divulged directly to users' social connections [19]. Two, the terms of service of many OSNs grant service providers rights to users' data. For example, these rights commonly include a license to display and distribute all content posted by users in any way the provider sees fit [9][11].

Recent work [5] [8], including our own [18], has proposed the use of decentralized OSNs as more privacy-preserving alternatives to the prevailing model. Decentralized OSNs distribute users' personal data across multiple administrative domains and thus reduce the chance of large-scale privacy breaches. In addition, they operate in a peer-to-peer fashion that gives users more control over their own data.

Decentralized OSNs present tradeoffs that bear further study. In particular, most centralized OSNs are free to users. Free services have obvious appeal, but they give rise to commercial pressures on providers to share users' data in ways that may diminish user privacy. In contrast, peer-to-peer OSNs generally require that users pay for some of the computing resources they use. We feel it is important to explore alternatives to the OSN status quo even though these alternatives may cost money to users, especially as popular awareness of privacy issues grows and the price of computing drops.

In this paper, we present three architectural alternatives for decentralized OSNs and compare their privacy, cost, and availability tradeoffs. Those alternatives are:

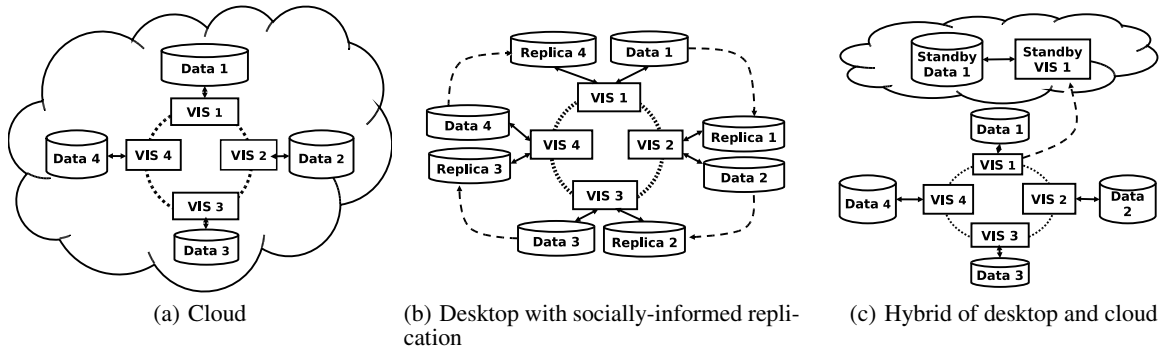


Figure 1: Alternative approaches to decentralized OSNs. In all three, users store their own personal data in their own Virtual Individual Servers (VISs), which communicate in a peer-to-peer fashion. The approaches differ in where VISs and data reside, which presents privacy, cost, and availability tradeoffs.

- Cloud-based decentralization.
- Desktop-based decentralization with socially-informed replication of user data.
- A hybrid of cloud- and desktop-based decentralization.

Figure 1 shows a visual representation of the three architectures. In all three, each user stores her own personal data in her own *Virtual Individual Server (VIS)*. VISs self-organize into peer-to-peer overlay networks, one overlay per social group with which the VIS owner wishes to share information. This structure supports many of the social networking features provided by popular OSNs, such as forming groups, finding friends, exchanging messages, etc. We use the word “virtual” because VISs can take the form of virtual machines, which offer important manageability and other advantages when compared to physical machines.

The three architectures differ in the placement of VISs and personal data. In cloud-based decentralization, VISs are hosted by a utility computing infrastructure such as Amazon Elastic Compute Cloud (EC2) [2]. The main advantage of this scheme is its high availability. Unfortunately, hosting a VIS in a cloud is currently costly. Continuously running a virtual machine at EC2 costs upwards of US\$75 per month.

In desktop-based decentralization, VISs run on desktop-class machines owned by OSN users. This solution has lower cost, but suffers from lower availability due to the high churn of desktop machines [13]. To increase availability, we propose a novel socially-informed replication scheme. The main insight behind this scheme is that users may be willing to replicate some of their personal data on machines belonging to social connections who would in any case have access to the data through normal OSN operations.

Hybrid decentralization is a combination of desktop-based and cloud-based decentralization. During normal operation, a user’s data is served from a VIS running in the user’s own desktop machine. When this machine becomes unavailable, a standby VIS is resumed inside the utility computing infrastructure. This solution could combine high availability with low cost because the cloud-hosted VIS would provide a stable backup while remaining quiescent most of the time.

The rest of this paper discusses these three solutions in further detail. It also presents our ongoing implementation and evaluation efforts, and surveys related work.

2. CLOUD-BASED DECENTRALIZATION

In a recent paper, we proposed Vis-à-Vis [18], a decentralized approach to online social networking. In Vis-à-Vis, each person

stores his own data on a personal virtual machine instance called a Virtual Individual Server (VIS). VISs self-organize into structured overlay networks that represent OSN groups, with one overlay per group. The same VIS can belong to multiple overlay networks, just as one person can belong to multiple social groups. VISs communicate with each other as peers and are free to reside anywhere in the Internet.

In Vis-à-Vis, VISs run in the cloud, more specifically in a paid utility computing environment such as Amazon EC2 [2]. We believe that individual consumers will adopt virtualized utility computing for many of the same reasons that enterprises have: it unburdens them from maintaining their own high-availability hardware without forcing them to give up control of their data, software, and policies. In contrast to free OSN services, paid utility computing services allow users to retain full rights to the content and applications that users place on these services [4].

Vis-à-Vis is based on a two-tier Distributed Hash Table (DHT) structure composed of a set of highly available VISs. The top-tier DHT, called the Meta Group, is used to advertise and search for public OSN groups. The lower-tier DHTs correspond to OSN groups and are maintained by the VISs of the group members. This requires VISs to maintain routing state for the Meta Group and every group of which they are members. Figure 2 shows a Vis-à-Vis network of eight VISs and three groups. Group 1 is composed of VISs A, B, E, and H; Group 2 is composed of A, C, D, F, and H; and Group 3 is composed of B, G, and H. All VISs are members of the Meta Group.

Vis-à-Vis offers a number of attractive features:

- It supports open and restricted groups.
- It supports public and secret groups.
- It scales to both very small and very large groups.
- It supports common OSN operations such as finding groups of interest, joining a group, leaving a group, and searching for information within a group.

Experimental results using our prototype implementation indicate that Vis-à-Vis is a viable alternative to the centralized OSN architecture. The latency of common OSN operations grows slowly, if at all, with the size of the corresponding OSN group. Similarly, the memory required by a VIS to participate in Vis-à-Vis is manageable and grows with the size and number of OSN groups to which a user belongs. It is important to note that the highly available nature of cloud-hosted VISs has the potential to mitigate some of the scalability problems seen in more general DHT deployments. For example, the frequency of DHT maintenance messages can be kept low since nodes can be expected to fail rarely.

