Key Differences between Web1.0 and Web2.0

Graham Cormode and Balachander Krishnamurthy
AT&T Labs–Research
180 Park Avenue, Florham Park, NJ {graham,bala}@research.att.com

February 13, 2008

Abstract

Web 2.0 is a buzzword introduced in 2003/04 which is commonly used to encompass various novel phenomena on the World Wide Web. Although largely a marketing term, some of the key attributes associated with Web 2.0 include the growth of social networks, bi-directional communication, various 'glue' technologies, and significant diversity in content types. We are not aware of a technical comparison between Web 1.0 and 2.0. While most of Web 2.0 runs on the same substrate as 1.0, there are some key differences. We capture those differences and their implications for technical work in this space. Our goal is to identify the primary differences leading to the properties of interest in 2.0 to be characterized. We identify novel challenges due to the different structures of Web 2.0 sites, richer methods of user interaction, new technologies, and fundamentally different philosophy. Although a significant amount of past work can be reapplied, some critical thinking is needed for the networking community to analyze the challenges of this new and rapidly evolving environment.

1 Introduction

"Web 2.0" captures a combination of innovations on the Web in recent years. A precise definition is elusive and many sites are hard to categorize with the binary label "Web 1.0" or "Web 2.0". But there is a clear separation between a set of highly popular Web 2.0 sites such as Facebook and YouTube, and the "old Web". These separations are visible when projected onto a variety of axes, such as technological (scripting and presentation technologies used to render the site and allow

user interaction); structural (purpose and layout of the site); and sociological (notions of friends and groups).

These shifts collectively have implications for researchers seeking to model, measure, and predict aspects of these sites. Some methodologies which have grown up around the Web no longer apply here. We briefly describe the world of Web 2.0 and enumerate the key differences and new questions to be addressed. We discuss specific problems for the networking research community to tackle. We also try to extrapolate the current trends and predict future directions. Our intended audience consists of technical readers familiar with some of the basic properties of the Web and its measurement, and who seek to understand the new challenges presented by recent shifts in Web technology and philosophy.

At the outset we need to distinguish between the concepts of Web 2.0 and social networks. Web 2.0 is both a platform on which innovative technologies have been built and a space where users are treated as first class objects. The platform sense consist of various new technologies (mashups, AJAX, user comments) on which a variety of popular social networks such as Facebook, MySpace etc. have been built (we adopt the convention of referring to sites by name when their URL can be formed by appending .com to the name). Inter alia, in all these social networks participants are as important as the content they upload and share with others.

However, the essential difference between Web 1.0 and Web 2.0 is that content creators were few in Web 1.0 with the vast majority of users simply acting as consumers of content, while *any* participant can be a content creator in Web 2.0 and numerous technological aids have been created to maximize the potential for content creation. The democratic nature of Web 2.0 is exemplified by creations of large number of niche groups (collections of friends) who can exchange content of any kind (text, audio, video) and tag, comment, and link to both intra-group and extra-group "pages". A popular innovation in Web 2.0 is "mashups", which combine or render content in novel forms. For example, street addresses present in a classified advertisement database are linked with a map Web site to visualize the locations. Such cross-site linkage captures the generic concept of creating additional links between records of any semi-structured database with another database.

There is a significant shift in Internet traffic as a result of a dramatic increase in the usage of Web 2.0 sites. Most of the nearly half a billion users of online social networks continue to use Web 1.0 sites. However, there is an increasing trend in trying to fence social network user traffic to

stay within the hosting sites. Intra-social network communication traffic (instant messages, email, writing on shared boards etc.) stay entirely within the network and this has significant impact on the ability to measure such traffic from without. There is also a potential for "balkanization" of users as the key reason to join a particular online social network is the presence of one's friends. If a subset of friends are not present there then communication across social networks would be needed but currently this is not a feature. Such balkanization impacts future applications such as a search engine that could span social networks.

We do not study social aspects of how users' interaction with each other in real life might change as a result of online social networks. Nor do we speculate on the lifetimes of some of the currently popular Web 2.0 applications; for example, the constant stream of short messages that are sent to interested participants detailing minutiae of daily life. Instead we concentrate on technical issues and how work done earlier in Web 1.0 can benefit the ongoing work in Web 2.0. At least one important aspect—user privacy—is left for future analysis.

Contributions. The contributions of this paper are as follows:

- We describe the tell-tale features of Web 2.0 and highlight the broad differences between Web 2.0 and Web 1.0. We illustrate this with a detailed case analysis, where we evaluate a number of websites and show which of the observable features they exhibit that make them either Web 1.0 or Web 2.0.

- We describe issues of structure of Web 2.0 sites, which tend to resemble social networks more than the hierarchical model of Web 1.0. We pose challenges of connecting users across multiple sites, and measuring the impact and scope of group membership. We identify site features that lead to 'stickiness', and formulate problems of measuring this adhesion. We discuss connections across sites, in the form of 'para'sites which provide additional functionality for specific hosts, and through embeddings and mashups.

- We identify new problems of measurement in Web 2.0, specifically related to the new models of interaction given by: clicking, connecting, commenting, and content creation. Each of these requires new techniques to measure. We also describe the challenging of crawling and scraping Web 2.0, and to build tools and new techniques to help this data collection.

- We cover technical issues such as performance and latency, and the prospect of flash crowds in Web 2.0, not just in the traffic flood sense, but also as floods of comments and links. The

user-created content common to Web 2.0 creates new distributions of access patterns, leading to re-evaluations of the value of object caching.

- We conclude by looking beyond Web 2.0 to connections to P2P, and examine future trends.

2 What is Web 2.0?

"Web 2.0" is a term that is used to denote several different concepts: Web sites based on a particular set of technologies such as AJAX; Web sites which incorporate a strong social component, involving user profiles, friend links; Web sites which encourage user-generated content in the form of text, video, and photo postings along with comments, tags, and ratings; or just Web sites that have gained popularity in recent years and are subject to fevered speculations about valuations and IPO prospects. Nevertheless, these various categories have significant intersections, and so it is meaningful to talk broadly about the class of Web 2.0 sites without excessive ambiguity about which definition is being used (from now on, we use Web2 and Web1 respectively for brevity).

Deciding whether a given site is considered Web2 or Web1 can be a difficult proposition. This is not least because sites are dynamic, rolling out new features or entire redesigns at will, without the active participation of their users. In particular, there is no explicit version number and active upgrade process as there is with a piece of software or a communication protocol, and many sites are referred to as being in "permanent beta". Some sites are easy to classify[1]: social networking sites such as Facebook and MySpace are often held up as prototypical examples of Web2, primarily due to their social networking aspects which include the user as a first-class object, but also due to their use of new user interface technologies (Facebook in particular). Other sites are resolutely Web1 in their approach: Craigslist, for example, emulates an email list server, and has no public user profiles, or fancy dynamic pages.

Many sites are hard to categorize strictly as Web1 or Web2. For example, Amazon.com launched in the mid-1990s and has gradually added features over time. The principal content (product descriptions) is curated rather than user-created, but much of the value is added by reviews and ratings submitted by users. Profiles of users do exist, but social features such as friend links, although present, are not widely adopted. Each product has a wiki page associated with it,

[1] Our discussion is based on the structure of sites at the time of writing, Fall 2007, unless otherwise specified.

Feature class	Feature	Facebook	YouTube	Flickr	LiveJournal	MySpace	Digg	Friendster	Amazon	Ebay	Craigslist	Slashdot
Profile details	Age	✓	✓		✓	✓		✓				
	Location	✓	✓		✓	✓	✓	✓	✓	✓		
	Gender	✓				✓		✓				
	Testimonials	✓	✓	✓		✓		✓		✓		
	Other data	✓	✓	✓	✓	✓	✓	✓				✓
Connectivity	Friends	✓	✓	✓	✓	✓	✓	✓	✓			✓
	Subscriptions		✓	✓	✓		✓					
	Groups	✓		✓	✓	✓		✓				
Content	Main content	profiles	videos	photos	blogs	profiles	links	profiles	products	products	ads	articles
	Other content	photos			(photos)	blogs, video photos		photos	photos	photos		
	Tagging	✓	✓[+]	✓	✓[+]				✓			
	Friends only	✓		✓	✓	✓						
	Comments	✓	✓	✓	✓	✓	✓		✓	✓		✓
	Editable content								✓			
	Rateable content		✓	✓			✓		✓	✓		✓
	Viewing Statistics		✓	✓			✓			✓		
Technical	Public API	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	Embedding allowed	✓			✓	✓						
	Many RSS feeds		✓	✓	✓	✓	✓				✓	
	Private messages	✓	✓	✓	✓	✓		✓				

[+] Only content creators are allowed to assign tags.

Table 1: Table showing features of some popular Web sites

but these are little used. Other sites also contain a mixture of the old and the new; we focus our discussion on the new aspects.

Another heuristic to aid distinguishing Web2 and Web1 can be based on time: the term "Web 2.0" was coined around 2004, and many of the first truly Web2 sites began emerging in late 2003 and early 2004. So sites which have changed little in structure since the early 2000's or before may safely be considered Web1 (such as IMDB). A definition of Web2 by O'Reilly in 2005 [23] emphasizes Web2 as viewing the Web as a platform. It is fair to say that many of the ideas that are now called Web2 were seen in earlier forms in the efforts of AOL and Geocities. While AOL brought the Internet to the masses, it also emphasized the notion of contained communities within which people could interact. Geocities initially operated with an enforced metaphor of 'neighborhoods'. These are precursors of current notions of groups and communities finding new and larger audiences in Web2. However, most Web2 sites differ by more forcefully making the user a first class object in their systems, and by employing new technology to make interaction easier for the user.

Some of the important *site features* that mark out a Web2 site include the following:

- Users as first class entities in the system, with prominent profile pages, including such features as: age, sex, location, testimonials, or comments about the user by other users.
- The ability to form connections between users, via links to other users who are "friends", membership in "groups" of various kinds, and subscriptions or RSS feeds of "updates" from other users
- The ability to post content in many forms: photos, videos, blogs, comments and ratings on other users' content, tagging of own or others' content, and some ability to control privacy and sharing.
- Other more technical features, including a public API to allow third-party enhancements and "mash-ups", and embedding of various rich content types (e.g. Flash videos), and communication with other users through internal email or IM systems.

Some additional explanation is required. Testimonials are comments from other users posted directly on a users profile. These can be general approbation (as in Flickr), or more for chatting in public (Facebook's "wall"). These are common in Web2 but missing in the less user-centric Web1. Other data can often be added on the user's profile page: in Web2 this is information such

as job, favorite music, education etc., whereas in Web1 this is more often contact details (email addresses). Our category of subscriptions means the ability to "subscribe" to a feed of news or updates from select other users; this is handled internally, in contrast to RSS feeds which are publicly visible. Some sites offer many RSS feeds, per-user/group, whereas others like Slashdot only have feeds for a handful of broad categories. In contrast to this public sharing of information, 'Friends only' means that ability to make some or all information visible only to "friend" users. One can quickly verify that a site such as Facebook provides many of the above features, whereas Craigslist provides few, with many being inapplicable.

On the technical side, some of the common presentation technologies associated with Web2 sites include AJAX (autonomous Javascript and XML), in particular use of XMLHttpRequest to dynamically update a page without explicit reload actions; embedded flash objects e.g. to play music or videos without additional browser plug-ins. Likewise, we have not discussed issues like the ability to "remix" or mash-up content, embed, reference or annotate; we will mention all these issues in later sections.

Analysis of Popular Sites. A set of examples are analyzed in Table 1, based on the above list of site features. The first five listed (Facebook, YouTube, Flickr, LiveJournal and MySpace) are Web 2.0 sites, while Slashdot and Craigslist are Web 1.0. Amazon, Digg, Ebay and Friendster fall in between. These assignments are debatable: Friendster seems to have many of the 'social' features in common with Facebook, but we consider it 'Web 1.5' since it fails to offer sufficient ways for users to interact with the content.

Although much touted, features such as the ability to collaboratively edit content (i.e. Wikis) are insignificant amongst the sites considered here. The notion of "tagging" is widely discussed, but only Flickr and, to a lesser degree, Facebook and Amazon, support tagging of other people's content; in other cases, tagging is limited to the content creator assigning tags to their content. Assigning ratings to content, or seeing statistics such as number of views is also far from ubiquitous. "Social" features, such as identifying friends, are integral to many of the Web2 sites. These are present on other sites, but less prevalent: Amazon does have friends, but this feature seems little used; Slashdot also has friends, primarily to adjust the importance of comments submitted by friends. These sites are quite functional if no friends are listed. In contrast, Facebook requires the user to add friends in order to to access most of its functionality.

3 Analysis issues

We now examine the various analytical properties of interest in Web 2.0 and contrast them against the properties that have been studied extensively in Web 1.0. These properties deal with how the Web2 sites interact with individual users. Some are relatively new and do not have a Web1 counterpart but many do, and we can compare the methodologies used to study them earlier in Web1.

3.1 Site Structure

Studies of Web1 highlighted a distinctive 'bow-tie' structure [4], with three distinct pieces of a massive connected component. Individual sites typically adopted an approximately hierarchical structure, with a front page leading to various subpages, augmented by cross-links and search functions. Web2 sites are often more akin to real-world social networks [20], which show somewhat different structures, due in part to implicit bi-directionality of links. There are some tattered remains of a bow-tie still visible [16]. Studying a Web2 site in detail can be inherently harder than studying the Web1 ecosystem, since it requires crawling *deep inside* the particular Web2 site. Some sites enforce a very user-centric view of the site, meaning that each account can only see detailed information about explicit 'friends' (see e.g. Facebook and other examples detailed in Table 1), in comparison to Web1 which is typically stateless. In particular, the trend is towards an increasingly customized 'front page' so that no two users have the same experience. In the Web1 case the crawling could be done externally without a login using a generic crawler. Increased use of a variety of server-side and browser-side technologies, in particular Javascript, can give further challenges for crawling Web2 sites.

The nature of a 'page' in a Web2 site is different from a Web1 site and the rate of change is likely to be significantly different due to increased interactive features. In commercial Web1 sites, content is centrally updated at somewhat predictable intervals. Individual users edit Web1 sites at differing frequencies. An early study [10] showed that many resources were not modified while some were modified quite frequently. There was a direct correlation between the popularity of a site and its rate of change: popular sites tend to change frequently. In Web2, with a lot of user generated content, it is not uncommon to have small incremental additions to the site. The

changes do not have to be done by the content 'owner'—friends can write comments (e.g., on their Facebook 'wall') which would constitute a change. A page is more a shared space in Web2 while in Web1 it is often a single-user writing medium.

Web2 often involves dynamically generated pages from multiple sources of information. It is thus harder to come up with a clean definition of a resource and determine when the resource has changed. This has implications on how often contents in Web2 need to be re-examined, how frequently could contents be fetched by a crawler, as well as implications on any caching (examined further in Section 6.3). A Web2 site is live in the sense that it can be updated while a user is examining it. The content within a Web2 page is a broader mixture of audio, video, text, and images, compared to typical Web1 site. The content types have additional implication on the rate of change but are probably similar to Web1. A more frequently changing entity on a Web2 page might be links to friends etc. which does not have a Web1 counterpart.

Recent work is starting to study the underlying "graph" structure of the social networks embedded in Web2 sites. So far, these typically look at a single site and measure properties such as degree distribution, clustering coefficient, connected components and so on. Initial work has plotted the degree distributions of various social networking sites, fitted them to power laws, and explored other properties of the induced graphs [21]. There will be interesting differences between sampling needed to compare within Web2 sites as opposed to the link structure in Web1 sites. It is a given that there will be many Web2 sites that have intra-linkage but blogs (Web 1.5?) don't fall in this category. Prior analysis shows that there are a lot of links to other blogs and non-blog sites [3].

Issues. This leads to many questions about how individuals use Web2 sites which simply do not arise in the Web1 world. For example, do users "live" on one site, or are they spread across multiple sites? This is hard to quantify from the researcher's perspective, since they do not have access to logs from any of the sites. Site owners may only be able to find some information via third party aggregators (e.g. outsourced advertisers can match up user visits). But there is likely to be very little outsourced from a Web2 site and thus third party aggregation may not have much resonance. This is a key difference between Web1 and Web2. Instead, one can look for explicit application level indicators of the same user across multiple sites—instances of the same username, profile links between sites, syndication of content (e.g. Flickr streams) and so on.

Other questions that arise include:

- Can we match individual users across multiple sites (and hence learn more of their attributes)? This is a more challenging exercise requiring more machine-learning techniques, bringing with it a higher potential for false-positives.
- Does the same user on different sites show the same behavior/connectivity? This seems much harder than simply determining if it is the "same" user.
- Given that a user is on one site, what is the probability that they are in another (affinity)? On Web1 visitors who visited CNN may visit NYTimes but a user may typically spend examining photos regularly on a single photo site on Web2. But the amount of time spent on Web2 sites may differ as the reasons are more social while the content is the focus in Web1 sites.

3.2 Advanced Structure

In Web1, all links and pages can be treated essentially equally, whereas trying to understand a Web2 site in detail requires looking at different link types (friend links, navigation links etc.) and page types (user pages, content pages etc.), which are rarely explicitly marked as such in a machine-readable fashion.

Other structures are often present on Web2 sites beyond generalized links, such as groups, subscriptions to feeds or message streams. These generalize and enrich features offered by Web1 sites which were originally just glorified emailing lists (egroups/Yahoo! groups), and make them more integrated parts of a Web2 site. The importance of such features is still to be determined. Many Web2 sites have no notion of groups. A common way in which the group feature is used is simply to make additional statements about the individual user—people join groups to express views in support of or against politicians, site features, movies, activities, etc. [2].

Issues. Similar to links, many natural measurement questions arise:

- How widely used are groups, and how active are they?
- What is the distribution of group membership (power laws, size distribution, membership distribution)? What is the distribution of *duration* of membership (joining to leaving)?
- How important are groups, subscriptions etc. in engaging users?

[2]There is even an (utterly boring) SIGCOMM Facebook group.

Such questions do not seem to have arisen or attracted much study in Web1 (although there is some analysis of usenet groups [26]). Yet these questions affect our understanding of how to provision and serve Web2 sites.

3.3 Site Mechanisms and Incentives

A key difference in Web2 is that many sites encourage users to spend as much time as possible on their site. There are strong incentives for increasing such *stickiness*: opportunities for higher advertising revenue. Further, the more the users interact with the site, the more can be learnt about their interests and habits.

In Web1 most sites have links to external sites and users may easily follow links to other sites. The main reason for this is that most Web1 sites tend to cover a single topic and do not require users to log in to access them. Web2 sites promote intra-site activities, often requiring users to log in and build links to others on the site. When users have logged in, sites can more easily track individual's browsing habits, and serve up personalized content. Users are encouraged to create an account in order to more fully engage with the site—some sites require accounts to post comments, others require accounts before *any* content is visible. Navigation links are often directed solely within the site, and where user content is allowed, external links may be made difficult or impossible to add. The mix of content in a Web2 site is typically more diverse than a Web1 site, reflecting the mix of interests of their user base, and increasing the probability of users to stick around the site. Web1 sites that do not allow user participation in a visible manner can only compete on the basis of content. User generated content even in the form of comments have been rare until recently on Web1 sites.

Explicit attempts to create stickiness for a Web2 site lead to 'portalization': trying to build every possible feature into the site, where once the user signs in, they never need to leave. This echoes the attempt of Web1 sites to become portals, with many features (news, weather, sports) accessible from a single front page. However, Web2 instead relies mostly on its users to bring content. Web2 examples of this trend include MySpace, which now provides hosting for users' photos and videos, and has intermittently blocked external content from being included on MySpace pages; and the opening of the Facebook API, which allows many features to be added to users pages, all within the Facebook domain.

Such portalization leads to a large amount of duplication of features: almost every Web2 site gives its users an 'inbox', essentially creating an internal email system which recalls the pre-Internet world of many non-interoperable local email systems. In order to ensure that users see their messages, often a (standard) email alert is sent to the registered email address of the user alerting them to the fact that a new message has arrived in their (Web2) inbox: Table 1 shows that this is common in most Web2 sites. Other sites are creating their own parallel Instant Messaging networks, allowing pairs of online users of the site to chat through their browsers, etc.

Issues. The following questions may need some careful modeling or innovative measurement studies to address:

- Will portalization efforts succeed? What is the equilibrium state as new ideas emerge? How many inboxes can one person cope with? Will the rate of active interactions with their inboxes vary across different social networks?
- What are the various technologies that can go inside the portal (IM, VoIP, P2P)? In other words, will all/most of a member's interaction with others be done through or as part of the social network, and how much through 'interoperable' email? Will such distinctions erode due to open standards?
- What are the non-economic incentives which keep users coming back to the same sites (e.g. casual games, announcements, active presence of friends, status updates)? How can the effectiveness of such incentives be measured?

4 Web 2.0 substrate and enabling technologies

We next provide an overview of the underlying communication model and technologies in Web2. Since users are first level objects in Web2 they are both producers and consumers of content. The role of the Web2 substrate is to help in the production of such content, host it, and allow interested users to consume it while interacting with other like-minded users.

4.1 Web 2.0 viewed as Publish/Subscribe model

We consider the ways in which Web2 sites move content between creators and consumers. Web2 widens ways to view content: on the website associated with the publishing site, syndicated to

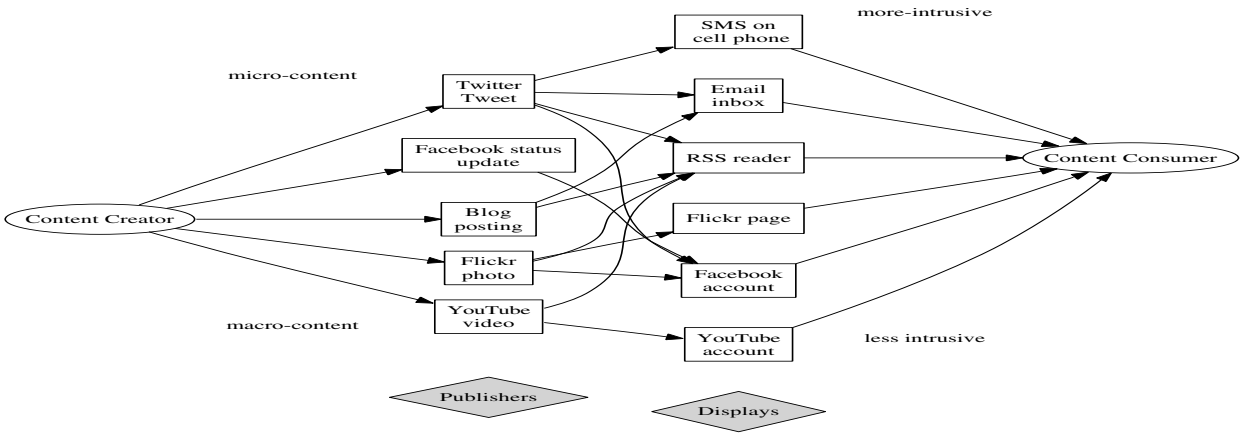

Figure 1: Paths from content creator to consumer in Web2

other sites, aggregated to RSS readers and email, and short content or alerts directed to cell phones. Figure 1 shows some of the possible pathways from users who create content through "publishers" to users who subscribe to content and view it on "displays". The data travels from publishers to displays by a mixture of push and pull: Twitter (which supports publishing status updates of up to 140 characters) can push these to a cell phone as an SMS, or have the content pulled by an RSS reader.

There is no inherent technological reason why any particular connection from a publisher to a display is not currently possible. Yet, many pairings are not currently supported, and others are addressed on an ad-hoc pairwise basis: many applications are being written solely to move content from publishers into Facebook using public APIs, for example. An obvious challenge is to encourage common interchange formats and protocols to eliminate redundant work. RSS is a candidate, but is limited since it does not allow authentication (some transfers should only be allowed to authorized subscribers). In some cases, publishers restrict which displays their content is permitted on, via artificial barriers.

Some routes from publishers to displays involve intermediate steps, such as gateways for SMS to email. Others routes create mash-ups by combining information from multiple sources, via open APIs, scraping, and structured information (RSS and other XML sources). This notion simply does not exist in Web1, and reasoning about such dynamic objects in Web1 terminology makes little sense, since they are not hosted or owned by any one site. Note that Table 1 shows that most

popular Web2 offer a powerful open API to address their content.

Issues. It remains to study the pub/sub view of Web2, and analyze the extent to which information can be channeled from publishers to displays, and which routes are blocked. Various sites are more or less permissive of what kinds of objects (e.g. Javascript or flash) are allowed to be embedded in their pages, and it seems inevitable that the large number of possible interactions will lead to interesting security holes in the vein of cross-site scripting.

4.2 Web 2.0 as a platform

A recent trend, driven by Facebook, is to view Web 2.0 as a platform supporting other applications. This is enabled by the opening of APIs and allowing users to add applications to their account, and share some information (such as their neighbors in a social graph) with the application. This development is quite recent (Facebook launched this feature in May 2007) and so there has been little formal study, and only a few months of history in comparison to the fifteen plus years of history of the Web as a whole.

The simplest enhancements allow a users to include content from a large variety of sources, which is often explicitly encouraged by the publishers: each YouTube video page (by default) includes the code required to embed the video into another page. Because of this flexibility, we start to observe a new class of site, a "para" site which provides services and features designed around a single host site. For example, many sites offer page designs, layouts, and other graphical embellishments for MySpace (e.g. WhateverLife [25], Pimp-My-Profile). In Web1, many sites offered generic enhancements—the one-time ubiquitous "Web counter"—which were suitable for adding to any page; in contrast, these para sites offer functionality only for a single targeted host at a time.

Richer applications make more extensive use of more recently opened APIs, and several of the most successful applications seem to derive from the para sites mentioned above. Within Facebook, the current most popular applications are provided by RockYou and Slide, which add additional "flair" to profiles in the form of photo slide shows and embedded video, and extend the capabilities for user interaction via posting richer messages and drawings, and giving virtual gifts. Such applications are somewhat akin to Firefox extensions, in that they may be displayed in a central repository with a tepid endorsement, but the actual application is maintained and executed

at the external site. A Facebook application in a user's profile is rendered by calls from Facebook to the application hosting site. The external site can thus get accurate metrics of usage but the access information is split between the various external sites and the host which provided the distribution channel and enabled the downloading. Unlike Firefox extensions, where communication is mostly local to the user's browser once the extension has been downloaded, external applications trigger a considerable amount of intra-site traffic in Web2.

The benefit to placing an application within a Web2 site with a social networking component (compared to directly hosting it on the Web) is the ability to leverage the existing network of friends of the user, and grow in popularity by viral spread. The disadvantage is that applications are at the mercy of the host, which can change its API or acceptable use policy at any moment, and can block any applications at will. Applications also compete for the scarce resource of screen real estate: installed applications or other plugins are typically shown one after another on the user's profile page.

Issues. Just as with adoption of Web2 sites as a whole, the spread of application usage within Web2 sites is open to study along with factors affecting the speed of infection, and the duration of popularity. With the possibility of such viral spread comes the possibility of flash crowds which may knock over the application servers, while the hosting platform remains up and running. We have to model the ecosystem of sites and applications/para sites, and understand what populations evolve. A more fundamental question is whether such applications represent a quantum leap in the evolution of Web2, or merely a brief fad. Do these add-ins represent missing functionality from the host site, and what happens when the host site adds this functionality to its core set of functions?

4.3 Key underlying Web 2.0 technologies

AJAX stands for asynchronous Javascript and XML, and is one of the key visible building box in popular Web 2.0 technologies. Ajax is a mixture of several technologies that integrate Web page presentation, interactive data exchange between client and server, client side scripts, and asynchronous update of server response. The Ajax intermediary sits on the client side sending requests to a server and updating the page asynchronously. A key component of the open standards-based AJAX is the Application Programmer Interface called XMLHttpRequest (XHR) that scripting languages use to exchange data between a client and a Web server. The data is *often* in XML format

but can be HTML, text, Javascript arrays, or even a few customized formats. Likewise, the scripting language does *not* have to be Javascript. XHR is not a protocol extension and was not introduced in any formal manner, rather, a feature of Microsoft's ActiveX was extended to other platforms.

The key purpose of Ajax is to let scripts act as HTTP (or HTTPS) clients and send/receive data from Web servers using a variety of common HTTP methods (GET, HEAD, POST, PUT, DELETE, and OPTIONS are supported currently). Thus, Ajax can be used for dynamic layout and reformatting of a Web page, reduce the amount of reloading needed by sending a request for just a small portion, and interact on demand with the server. The responses from the server are handled asynchronously by the browser without having to keep the user's attention frozen. Numerous popular dynamic Web applications such as maps use XHR.

Similarly, Flash objects can offer similar functionality in that once downloaded they can communicate asynchronously with a server. Consequently, YouTube videos can begin playing before the whole movie has been received: the user downloads a compact flash object which downloads a small prefix of the video and begins playing it out while asynchronously fetching the remainder of the video. Supporting Flash requires an appropriate Adobe plug-in to be installed, although user penetration of this plug-in is in the high ninety percentiles. Toolkits exist allowing Internet applications to be written in a high level language and then rendered either as Flash objects or pages with Ajax components, meaning that it may be helpful to sometimes think of Flash and Ajax merely as object code. Ajax apps are typically easier for the researcher to reverse-engineer and understand for measurement purposes than the Shockwave Flash (SWF) format. Currently, Flash is mostly used for rendering rich embedded objects (video, audio, games): few entire applications which store and recall data are implemented in Flash.

Issues. For both Ajax and Flash, it remains a challenge to develop tools and general techniques to be able to parse and analyze client/server communications. What are the implications of AJAX, asynchronous transfer etc., for servers? What is the distinction between the "auto-refresh" feature in Web1 whereby a site is automatically reloaded and the Web2 sites that contact a server to update part of the page on a regular basis.

5 Measurement issues

We examine how data can be collected from and about Web2 sites. Much can be learnt from prior work in Web1 and reused.

5.1 Traffic Measurements

In Web1 traffic measurement was based on precise, comparable metrics. The click count and page view defined quantities which could be measured through site traffic logs, and compared. More generally, Web1 measurements include popularity of sites, fraction of traffic on Internet, number of clients, servers, proxies (number of clients behind a proxy) and so on.

The shift in technologies that has accompanied the rise of Web2, in particular asynchronous transfers, has weakened the precision and comparability of these measures. A user can spend a significant amount of time interacting with a single page without ever triggering an explicit "page load" event (a 'click' in Web1 world): for example, consider a user scrolling and zooming in and out of an interactive map to plan a route. Thus there is a gradual shift to a less technologically driven metric in order to rate popularity (and hence set ad rates): measuring the amount of time a user spends on a site, instead of the number of discrete pages they view [2] and moving from pay-per-click advertising to pay-per-action [13]. This still leaves some uncertainty: just because a page is open in a tab of a browser, it does not mean it has the user's attention. Indeed, due to the asynchronous push based technology, users are incentivized to leave tabs open in the background, so that they can quickly scan the page later for updates (new messages, status updates, etc.).

The metric of hits on a Web site becomes problematic if a page sends out multiple XHR requests in Ajax, for small updates to the page. A site can easily inflate hit counts based on the micro requests and responses. Is the additional scripts or responses sent back as a result of a client side script that was invoked viewed as part of the contents of the page and made available to search engines? From a traffic point of view the number of HTTP requests are potentially larger as users trigger dynamic requests by interacting with the application. However, in many cases the requests can be in the form of a Javascript call that is handled locally at the client end avoiding a round trip to the server. If the requests are sent to the server, the responses are typically small and only a small portion of the page requires to be updated. If the user does not interact with certain parts of the

downloaded page, additional data/scripts need not be downloaded, thus reducing overall response size.

Given some metric, in Web2 it is still easy for a site to measure its own audience: all the necessary information is available to the site. However, to measure audience from outside requires different strategies. Moreover, Web2 is meant to shift its users from being passive viewers of Web1 content to being active creators of Web2 content. This brings new questions of measurement of activity which did not arise in Web1: how to measure the various actions of users? We consider the following classes of actions:

- "Clicks and connections": simple activities which only require a single click to complete, such as rating a movie, voting in a poll or voting for a story (as in Digg), or adding a semantic link, such as adding a friend.
- "Comments": adding a short response, comment or tag to existing content, such as a news story, blog entry, photo etc.
- "Casual communication": sending a message to another user, either via an email-like system or via instant messaging. These are typically short, a sentence or two per communication.
- "Communities": interacting in larger groups or communities by joining a group or posting a message to a group
- "Content Creation": uploading or entering some entirely new content, such as a webcam movie, digital photo, or blog posting.

For a given site, it is relevant to measure the frequency of these actions, and to measure what fraction of users participates in each one. Much of the hype surrounding Web2 focuses on content creation as being a key element. However, while many users have created and shared some content [19], on any given site new content is only created by a few users. Similarly, the most popular content may be dominated by a small clique (e.g. perceptions of bias in Digg front page [5]), or by professionally produced content (e.g. YouTube's most viewed list is dominated by music videos [29]). Other activities are also key to the success or failure of a Web2 site, and so it is also important to understand the less active interactions (comments, clicks and links), and also the virtually passive (views). Finding the relative distributions of these activities (distributions of comments per posting, votes per poll, ratings per movie) are key to understanding the more complex interactions. Simply taking a ratio of uploads to views for YouTube [30] gives a 1:1500 ratio of

active to passive. But this is overly simplistic, and does not capture the fact that a few videos get millions of views, while a large fraction (the long tail) receive only a handful. It also does not answer how many users of YouTube only ever view content.

Issues. In Web1 these questions do not arise, since the typical model is that all visitors are read-only, and all content and comment is provided by the site owners. In Web2 there is a commingling of commenters and creators, and every visitor has the opportunity to click, comment, create, etc. Extracting data on these actions is challenging, but possible, since the quantification and presentation of these actions is an expected part of the information delivered to the users: number of comments, number of views, number of ratings are all often presented publicly as metadata for each piece of content. Scraping these values currently requires significant effort, and is addressed in the next section. The following specific challenges present themselves:

- What is the correct metric for measuring Web2 traffic? How can this be accurately measured from outside a site?
- How to measure and calibrate the different levels of user interaction (clicks, comments, content creation) across different sites?

5.2 Crawling and Scraping

Underlying many of the issues discussed above is the need to be able to crawl a large Web2 site and the induced social network, and from the retrieved information extract rich metadata from the pages. As Web2 sites contain a variety of information, and each is structured in a different way, this typically requires building a crawler which is capable of parsing a page into different semantic elements (navigation links, friend links, group links, other links) in order to extract the social network and associated data. Some sites, such as Flickr and YouTube, offer APIs to extract comments, friends, views, and tags, which simplifies the extraction of data and links, but still requires appropriate custom code in order to allow large scale extraction of data. Recent presentation technologies (Javascript and XML) further complicate the crawling task, since in addition to parsing returned pages, crawlers also have to simulate user clicks in order to extract some additional data. Exploration by crawling can be seriously hindered if the site presents each logged in user only information about their reciprocated friends, since most crawlers have few friends. Finding

"backlinks" to nodes is also a challenge, especially if the inward-pointing nodes have few or no incoming links themselves [21]. For similar reasons, it is much harder to passively sniff such traffic and extract the pattern of user behavior, marking a move away from stateless interactions.

Thus far, studies have looked at individual or small collections of sites in isolation, such as YouTube, blog sites (LiveJournal, blogger), MySpace etc. [1, 12, 14, 15, 16, 18]. These give detailed views of certain popular sites, or some comparison between a couple of similar sites. Other studies have looked at wider settings, but only studying Web1 properties such as server properties, numbers of links, and not the additional Web2 semantics of friends etc., e.g. in studying collections of blogs [8]. Detailed, large scale comparison of sites currently requires significant effort in data collection, and so has not been realized. A more common approach is to look at use of specific Web2 sites traffic use at the edge of a campus network and obtaining a measure of popularity of resources [31].

YouTube in particular has attracted significant recent study. These studies have tried to analyze several aspects, such as the number of views and rankings, and local (geographic) popularity of videos over time [6, 31]; object sizes and access patterns [11]; and properties of the embedded social network such as degree and cluster coefficient [21]. Simply crawling a single large Web2 site such as YouTube brings out differences [11]: Web2 sites are a moving target with a significantly higher rate of change than popular Web1 sites. The volume of data to be fetched can be significantly high for a single Web2 site compared to that of the most popular Web1 site or even a collection of popular Web1 sites: CNN will only add a at most a few hours of of its video output per day, whereas YouTube has claimed over 65,000 new videos per day[3].

Issues. Professional-level crawling bandwidth would be needed to fetch even small portions of YouTube-like sites. In some cases this cost can be reduced by only "indexing" the site and its content, i.e. by only collecting statistics on large video and graphical objects, rather than fetching them. Indexing necessarily limits further analysis that can be done (e.g. analyzing bit-rate choices, other encoding features [11]). Finally, one expects that Web2 site owners such as YouTube should be protective of their bandwidth and the content stored on their sites, and so would prevent excessive crawling.

[3]This statistic originally appeared on YouTube's fact sheet page [30], but has since been removed. This highlights the need for independent methods of measuring activity.

- A challenge to the community is to start developing general purpose tools for crawling and parsing Web2 sites, which can be quickly customized for a particular site. Initial attempts to create such tools may have the side benefit of exposing commonalities across specific Web2 sites, and highlighting generic technical and data presentation issues.
- What techniques can be designed to probe "closed" sites such as Facebook, which only reveal information on friends of the user? One approach is to design plug-in applications via the API which collects (anonymized) data about users who use the application.

6 Technical and external issues

We now explore issues related to understanding and measuring Web2 sites as part of the Internet ecosystem. This is based on externalities imposed by user's presence and interaction with a site.

6.1 Performance and latency

This is probably the best studied aspect of Web1 and largely irrelevant to Web2. It is significantly easier to provision for in Web2 and most popular Web2 sites are significantly over-provisioned. Except for rare cases like cross-site scripting worm attacks, they do not experience significant daily latency fluctuation. Part of the reason is that the number of individual end users are largely fixed during small intervals of time and can only redirect their interest to different parts of the site. They can move from chatting to sending email to uploading audio/video; none of the individual actions can cause significant ripples on the overall network. However, external events such as a large number of users simultaneously applying operating system patches can have an impact. Many Web 2.0 sites impose a variety of restrictions on users and enforce them to ensure prevention of viral spreading of communication and data.

In Web2 there is considerable data about expected load based on the number of subscribed users. For example, Facebook's 55 million users provide a reasonable bound on the expected load. In Web1 the potential for flash crowd may be slightly higher. The key difference is that most social networks require a registration phase which effectively provides them with an upper bound on the number of users at any given time (and a control mechanism to slow down admission in the event

of a mass influx of new signups). Sites that do not mandate logging in, such as MySpace and YouTube, can experience sudden increases in load but use CDNs to relieve surges.

Flash crowds–where a large number of users simultaneously and unexpectedly try to access a Web site—have been documented since the mid-nineties. Popular Webcasts like that of Victoria's Secret company or special events like Olympics, popular sites such as Slashdot highlighting a site cause flash crowds directing a large amount of new traffic. A natural question is what are the equivalents in Web2, and how do they differ from a Web1 flash crowd. If they occur, what are their durations and distributions? Is there a single peak, or multiple waves?

A first observation is that a flash crowd in a large well-provisioned Web 2.0 system has no observable impact on the system performance. For example, the sudden popularity of a particular video clip on YouTube, or a band on MySpace will draw a lot of traffic to a specific page, but without significantly increasing the aggregate overall traffic to the site. Flash crowds consist of a large number of individual visitors. One can imagine the equivalents for different actions in Web2: a piece of content suddenly attracting a large number of ratings or comments, or a group gaining many new members. Again, since sites offering such facilities are typically large and over-provisioned, there is no technical reason why the impact on service should be visible. There may be reasons to try to 'fake' a large number of comments or ratings: to attract interest, or to drive traffic (directly, or via 'comment-spam' intended to raise search engine ranking of the target). This issue frequently arises in discussions of Digg, a user-voted "cool links" list, which can be considered a Web2 counterpart of Slashdot. Manipulation of voting is a concern, but has not been disruptive enough to lead to mass desertion of users. Successful manipulation appears to require exploits such as the MySpace worm that managed to generate a very large number of friends in a few minutes, or the involvement of site owner (for example, MySpace automatically adds founder Tom Anderson as a "friend" to all new accounts; Digg's creators initially removed all stories including an HD-DVD processing key).

Issues. The questions of interest here include, how new Web2 sites can scale with increasing popularity? Are there generic services that will help them to scale quickly while not increasing latency to all users? How can one independently measure popularity of applications within Web2 sites, e.g. Facebook or MySpace applications such as "Where I've been?". Although the host Web2 sites typically scale well with load, external embedded sites and application servers may not

cope so well, especially when their popularity spreads "virally" through the user community. Load can certainly be high, and skewed: 87% of usage of the thousands of Facebook applications go to just 84 applications [24]. Measuring and predicting these trends is a new challenge for Web2.

6.2 Configuration, Distribution, and Location

Software used for Web2 is quite different as the set of requirements and expectations are different. YouTube does not have the same load mix as CNN. Redirection rates may be somewhat similar but only with highly popular Web1 sites. Redirections are likely to be at the HTTP-level in Web2 rather than at a lower level in the protocol stack. If a particular Web2 site has wide client distribution and large load potential then a CDN might be used (e.g., the Joost video delivery service and YouTube's reliance on LimeLight). However, the decision to update content and possibly redistribute it is still carried out centrally.

It is easy to locate Web2 sites as they are fewer in number, more concentrated, with limited need for replication. For example, location-indicative subdomains such as `foo.myspace.com` and `bar.facebook.com` exist, (where `foo` is one of a number of countries, and `bar` one of many universities). Here, the intended location of the page is advertised, even if this does not correspond to the actual physical location. Given the "weight" of a Web2 site (much heavier than even popular Web1 sites) the bytes are expected to be closer to the users of the site and thus geographically constrained at country level. The differences between Web1 and Web2 are expected to be pronounced here. Tools used to narrow down locations in Web1 can be reused with a higher hit rate. Clients of a Web2 site are likely to be less-well spread out than a highly popular Web1 site, due to the emphasis on social aspects and linking to 'friends'.

Issues. Web2 compliance is entirely unstudied as are client connectivity issues. Given the significant differences in user demographics of Web2 sites (mostly younger with internal differences between MySpace and Facebook of a few years), there is an expectation of better connectivity for the more active younger users. Increasingly, however, mobile connections to Web2 sites bring up connectivity issues and ways to send data down a thin pipe similar to earlier work in Web1 (e.g., WebExpress, delta mechanisms [22] etc.). In Web1 alternate sites were created and content tailored to respond to devices that had bandwidth limitation. In the case of Web2, given the dynamism with which the site changes, sending content to mobile devices with constraints can be significantly

harder. However, short updates can be easily handled, e.g. as text messages.

6.3 User workload models

Reference patterns, object size distributions and so on are just starting to be studied [11, 21]. Initial studies indicate that size distributions approximately follow the familiar heavy tailed size distributions, even when limitations are enforced (e.g. most YouTube users are restricted to uploading 10 minutes worth of video). The manner in which data is gathered to carry out such studies shows an important difference: crawling across Web1 is much easier as the load imposed on a particular Web site is rather low and back-off strategies can be used; however crawling a heavy Web2 site will impose a significant load on that site and may have to be staggered over a much longer period. Early indications are that there is a spread of content types and the aggregation potential of individual content types' contribution to a Web2 site and potential caching impact remains to be studied. Studies [11] have found, for example, that video clips were longer on YouTube than the overall Web, and at higher bitrates.

Recent work [11, 31] examining popularity of YouTube in campus environments showed that local and global popularity of video clips are significantly different. Thus, proxy caching can be beneficial but modeling workload based on local conditions may be problematic. Different access patterns in Web2 vs. Web1 affect the efficacy of cache deployment: skewed distributions with long tails of object popularity means fewer cache hits, rendering caching to be not worthwhile. In the case of static video streams, it might make sense for ISPs to deploy caches if a significant number of videos are popular locally, as [31] indicated. However, if videos being streamed are modified with advertisements and constantly changing annotations, then caching becomes harder.

Although not solely a Web2 application, Instant Messaging (IM) shares characteristics with developing Web2 applications, which often offer IM capabilities within the browser (e.g. Meebo). A session in Web1 could be largely determined by examining connections between a client and a server site whereas in IM, there is a significant variance. This stems from different IM clients which impose different quiescent periods before a timeout. Protocol and 'keep alive' messages exceed chat traffic in IM [28].

Issues. The questions in the context of workload models include, is in-network caching of Web2 objects worth the effort and cost? This requires modeling growth in object size, object access

distributions, and disk vs. bandwidth cost to determine. It also requires predicting trends in access distributions: is the long tail nature of access getting fatter? How should one deal with newer applications such as Twitter, that have frequently updated micro-content, as compared to YouTube where macro-content is almost never updated, but much larger? How is the performance of a Web site measured in the presence of Ajax? Is there a contribution to visible latency, given that the updating is done asynchronously?

7 Summary of metrics of interest

Comparing the various metrics computed over the recent years in Web1 and what might be of interest in Web, we see some obvious overlap but there are also several new metrics of interest. We follow the set of issues raised in Chapter 7 of [9] to identify these metrics.

Web 1.0 metrics of similar relevance in Web2 include the overall share of Internet traffic, number of users and servers, and share of various protocols. Around half a billion users are present in few tens of social networks with the top few responsible for most of the users and thus traffic. These sites work hard to keep traffic within their own network via their own versions of email, instant message etc. Traffic inside a Web 2 is harder to measure without help from the site itself. For example, a user writing on a board ('wall') of a friend may result in notifications generated to other friends who have expressed interest. The notifications may only result in actual traffic when other friends log in, view the message and possibly respond. With a large fraction of users returning to the site more than once a day, there is bound to be considerable internal communication. However, such messages are short and human generated and are likely to be fairly bounded in overall bytes.

Growth patterns have been similar to some popular Web sites. Since there are registration requirements and a fairly quick drying up of close friends for each user, there is some tailing off effect. Almost all the popular Web2 sites are accessed over the Web implying HTTP and thus TCP connections. Facebook is the seventh most visited Web site (currently just behind Google). Traffic generated by some popular applications (such as Twitter) is mostly UDP, and the people requesting such notifications are pre-registered. If there is a steady increase of external feeds into the growing volume of users there could be an explosion in the number of connection setups. This will lead to pressure to streamline feeds to reduce the overhead much like persistent connections and

pipelining were introduced in HTTP/1.1. Given the control that individual Web2 sites have over their interface, streamlining can be much more rapid than the years taken for HTTP/1.1 adoption.

As discussed in Section 4.2, the set of external applications (almost 6,000 in Facebook alone) and widgets introduce a very different kind of challenge; one that is unique to Web2. Facebook has claimed that at least one application has been installed and used by virtually *every* user. Applications allow new interactions between friends, and trigger internal notifications after actions are taken (such as a move in a turn-based game). The overall traffic as a result of a Web2 site is thus the product of the set of interactive applications and the set of participating friends.

Web1 went through considerable effort to streamline popular sites for mobile users; in Web2 the challenges are slightly different. The fact that most communications are short and episodic allow for instant notification to users via mobile devices. Context is generally deprecated and any potential accompanying rich media can be deposited at the site for later perusal. Real-time requirements in Web1 only mattered for certain classes of Web sites (stock tickers and game score updates). In Web2, there are a class of communications (IM) that are time-immediate but writing on a user's board does not carry the same urgency. In fact, there is an explicit attempt to allow both kinds of communication in support applications: twitter for example allows the followers to get the notification ('tweet') on their board or on their cellphone.

In Web1 the communication between a client and a site is fairly limited and highly restrictive: a request is sent and the site responds. If there are too many requests some requests can be dropped and others can be delayed. A site can choose between classes of users. In Web2 since most communication is *between* users, the site has no easy way to select during overload. However, the sites can (and do) impose varying limitations to ensure that overall load and thus latency is maintained at a reasonable level. With increasing users any lack of scalability will result in additional restrictions. Decisions made by the Web2 site affect all users uniformly as there is no incentive in prioritizing classes of users.

8 Beyond Web 2.0

We conclude by going beyond Web 2.0, to draw connections to other application types, such as P2P and Skype, and analyze the impact of wider adoption of Web 2.0 paradigms.

Implicit Social Applications. Skype, which offers voice calling over the Internet, now has over 80M users globally, and is constantly adding new features (conference calls, voice mail). There is no reason not to think of it as a social network that allows people to exchange voice bits and text and form a community of interest (call list). As such, it can be modeled and viewed through the same lens as the Web2 sites which we present in this paper, and many of the same questions apply (some initial measurements are in [7]). It differs from other examples we have identified in the main content (voice conversations) and hence the volume of bits involved.

Peer-to-peer and Web 2.0. A P2P peer who supplies content of interest is not be a *friend* in the social networking sense. Friends in real life may share interests in similar content (books, music etc.) but often they share *pointers* in the form of recommendations. In the P2P sense, friends in real life act as .torrent files. There may well be interest in consuming the bits simultaneously and interacting as people do now over the phone while watching a sports event.

The Web 2.0 Electronic Fence. Web 2.0 can bring balkanization—people in one social network may not communicate frequently with some of their friends who spend more time on other social networks. Artificial separation into tribes is encouraged by some of the Web2 sites who want to maximize and retain the set of members inside their "electronic fence". However, there is a counter-current due to the prevalent link-based nature of the Web—users will constantly link to sites outside the fence. This will be sufficient to prevent complete balkanization.

Web2.0ification and sharing friends. More sites are inviting users to "add friends", but there are only so many times that a user wants to find which of their friends on the same site. If this is not necessary to use the site, then users can ignore this, or use a 'bugmenot' equivalent. But for some sites (such as Facebook), all value comes from connecting to friends. Sites currently offer the highly dubious (in terms of both security and accuracy) technique of users sharing their email addressbooks in order to find contacts via email address matching. One proposal is to allow users to record their "social graph" (encoded in XML formats such as FOAF) once, and allow different sites to access this information, essentially linking up all the currently isolated graphs (the MySpace graph, the Facebook graph, the Flickr graph). More insidiously, 3rd party sites can tap into a user's social connection via open APIs and cookie-sharing agreements with a Web 2.0 site acting as an identity manager, akin to a widened notion of the Microsoft Passport.

Privacy and Security. We iterate that there are significant challenges in allowing users to un-

derstand privacy implications and to easily express usage policies for their personal data. Privacy is typically not well understood by Web2 users, resulting in unintended consequences. Many teenagers accept that their posted data may unintentionally identify them [17]. Simultaneously, dynamic presentation technologies also raise security concerns [27]. Both privacy and security in Web2 demands explicit study and analysis.

Thanks. We thank Dave Kormann, Hal Purdy, Walter Willinger, and Craig Wills for their helpful comments.

References

[1] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *ACM SIGKDD*, 2006.

[2] S. Bausch. Nielsen//netratings adds "total minutes" metric. `http://www.nielsen-netratings.com/pr/pr_070710.pdf`, July 2007.

[3] S. Bhagat, G. Cormode, S. Muthukrishnan, I. Rozenbaum, and H. Xue. No blog is an island analyzing connections across information networks. In *International Conference on Weblogs and Social Media*, 2007.

[4] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, 2000.

[5] B. Carr. Is digg broken beyond repair? `http://www.thenewbusinessblog.com/miscellaneous/is-digg-broken-beyond-repair/`, Apr. 2007.

[6] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *ACM SIGCOMM IMC*, 2007.

[7] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying skype user satisfaction. In *ACM SIGCOMM*, 2006.

[8] E. Cohen and B. Krishnamurthy. A short walk in the blogistan. *Computer Networks*, 50(5):615–630, 2005.

[9] M. Crovella and B. Krishnamurthy. *Internet Measurement: Infrastructure, Traffic & Applications*. John Wiley and Sons, 2007.

[10] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul. Rate of change and other metrics: A live study of the World Wide Web. In *USENIX Symp. on Internet Technologies and Systems*, 1997.

[11] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube traffic characterization: A view from the edge. In *ACM SIGCOMM IMC*, 2007.

[12] L. Gu, P. Johns, T. M. Lento, and M. A. Smith. How do blog gardens grow? Language community correlates with network diffusion and adoption of blogging systems. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006.

[13] R. Kniaz. Pay-per-action beta test. `http://adwords.blogspot.com/2007/03/pay-per-action-beta-test.html`, Mar. 2007.

[14] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. Structure and evolution of blogspace. *Communications of the ACM*, 47(12):35–39, 2004.

[15] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW Conference*, 2005.

[16] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *ACM SIGKDD*, 2006.

[17] A. Lenhart and M. Madden. Teens, privacy & online social networks. Technical report, Pew Internet and American Life Project, Apr. 2007.

[18] T. Lento, H. T. Welser, and L. Gu. The ties that blog: Examining the relationship between social ties and continued participation in the wallop weblogging system. In *3rd Annual Workshop on the Weblogging Ecosystem*, 2006.

[19] M. Madden and S. Fox. Riding the waves of "web 2.0". Technical report, Pew Internet and American Life Project, Oct. 2006.

[20] S. Milgram. The small-world problem. *Psychology Today*, 1:61—67, 1967.

[21] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *ACM SIGCOMM IMC*, 2007.

[22] J. C. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy. Potential benefits of delta encoding and data compression for HTTP. In *ACM SIGCOMM*, 1997.

[23] T. O'Reilly. What is web 2.0. http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, 30th September 2005.

[24] T. O'Reilly. Good news, bad news about facebook application market: Long tail rules. http://radar.oreilly.com/archives/2007/10/facebook_long_tail_report.html, 5th October 2007.

[25] C. Salter. Girl power. http://www.fastcompany.com/magazine/118/girl-power.html, Sept. 2007.

[26] M. A. Smith. *Invisible Crowds in Cyberspace: Mapping the Social Structure of the Usenet*. Routledge Press, 1999.

[27] A. Stamos and Z. Lackey. Attacking AJAX Web Applications. http://www.isecpartners.com/files/iSEC-Attacking_AJAX_Applications.BH2006.pdf.

[28] Z. Xiao, L. Guo, and J. Tracey. Understanding instant messaging traffic characteristics. In *International Conference on Distributed Computing Systems (ICDCS)*, 2007.

[29] You tube most viewed (all time). http://www.youtube.com/browse?s=mp&t=a&c=0&l=.

[30] Youtube fact sheet (archive copy). http://web.archive.org/web/20070221115744/http://youtube.com/t/fact_sheet, Feb. 2007.

[31] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch global, cache local: Youtube network traces at a campus network - measurements and implications. In *IEEE MMCN*, 2008.