Exploiting Locality of Interest in Online Social Networks

Mike P. Wittie, Veljko Pejovic, Lara Deek, Kevin C. Almeroth, Ben Y. Zhao
Department of Computer Science
University of California, Santa Barbara
{mwittie, veljko, laradeek, almeroth, ravenben}@cs.ucsb.edu

ABSTRACT

Online Social Networks (OSN) are fun, popular, and socially significant. An integral part of their success is the immense size of their global user base. To provide a consistent service to all users, Facebook, the world's largest OSN, is heavily dependent on centralized U.S. data centers, which renders service outside of the U.S. sluggish and wasteful of Internet bandwidth. In this paper, we investigate the detailed causes of these two problems and identify mitigation opportunities. Because details of Facebook's service remain proprietary, we treat the OSN as a black box and reverse engineer its operation from publicly available traces. We find that contrary to current wisdom, OSN state is amenable to partitioning and that its fine grained distribution and processing can significantly improve performance without loss in service consistency. Through simulations of reconstructed Facebook traffic over measured Internet paths, we show that user requests can be processed 79% faster and use 91% less bandwidth. We conclude that the partitioning of OSN state is an attractive scaling strategy for Facebook and other OSN services.

1. INTRODUCTION

Online Social Networks (OSN) are fun, popular, and socially significant. Their core functionality is the archival and delivery of communications along social links between users. Integral to OSNs' success is the size of their global user base: Facebook has more than 400 million users, over 70% of whom are outside the U.S. [10]. To provide a consistent view of the social network to every user, Facebook relies on large data centers in the U.S. to host full replicas of the OSN's state. As a result, requests from users outside the U.S. suffer from slow response time, or lag, and put high load on the Internet backbone. In scaling its infrastructure by replication, Facebook has put an emphasis on consistency over performance. Could Facebook have achieved both goals all along?

In this paper, we investigate the causes of Facebook's poor performance and the opportunities for improvement. Judging by their non-trivial efforts to optimize communication protocols, Facebook is keenly interested in reducing delay and hosting costs [20, 26]. However, many details of Facebook's service remain proprietary, and so we treat the OSN as a black box and reverse engineer its traffic and operations from publicly available traces and custom network measurements.

Our analysis lets us make two observations. First, delay experienced by users outside the U.S. is due to the interplay between the Facebook communication protocol and Internet path characteristics. While the volume of an individual Facebook request is low, the number of involved round trips is high. Internet paths between hosts outside the U.S. and Facebook data centers are characterized by high latency and high loss. The combination of high latency and many round trips causes long request delay, further increased by retransmissions.

Second, the centralization of OSN servers in the U.S. means Internet bandwidth is wasted when many users in a region outside the U.S. request the same data. While Akamai solves that problem for static content, a significant share of regional traffic is dynamic and shuttled to the U.S., thereby wasting Internet bandwidth.

To mitigate the problem of high latency and high loss paths, we observe that these characteristics can be isolated by a well-connected server in each region, suggesting the use of TCP proxies, which have been shown to reduce TCP transfer duration in similar scenarios [4, 24]. To reduce the load of OSN traffic in the Internet, we observe that the majority of communications are between users within the same geographic region. This high *locality of interest* suggests that partitioning and distribution of the social graph required for processing and caching of regionally generated content could allow many requests to be processed in the region, thereby reducing the volume of OSN traffic in the Internet.

.

Based on these observations, we believe that a distribution of Facebook's architecture at a finer granularity has the potential to improve OSN service responsiveness and efficiency from the network perspective.

TCP proxies and caching are well understood techniques, however, their effectiveness depends on specific traffic and network characteristics. While it is intuitive that these solutions should be effective at improving OSN service responsiveness and efficiency, we quantify their effectiveness by simulating TCP connections of Facebook traffic reconstructed from their social graph and history of communications within a number of fast growing regional networks. This approach is novel in its focus on the interplay between user behavior, mechanisms of a black box OSN, and network characteristics.

We show that regional servers improve service quality and reduce the mean user request delay in Russia, for example, by 79% from 3.4 to 0.7 seconds. We also show that regional servers improve service efficiency and reduce data center load from Sweden, for example, by 91% from 568 Mbps to 49 Mbps and network load for the same interactions by 83%. These reductions in request delay and traffic load are statistically significant and meaningful improvements for Facebook users. We also show these gains can be achieved without sacrificing Facebook service consistency. Although the deployment of regional servers represents additional cost, their benefits and higher efficiency have the potential to offset the initial investment and indeed present a more cost effective scaling strategy than wholesale data center replication.

The rest of this paper is organized as follows. In the next section we describe the Facebook service through a reverse engineering effort. Section 3 describes the implementation of TCP proxies and OSN caches. Section 4 details the evaluation methodology we use to quantify the proposed architectural changes in Section 5. We summarize related research in Section 6 and conclude in Section 7.

2. REVERSE ENGINEERING FACEBOOK

Motivated by the size and social significance of Facebook we are interested in understanding the factors behind its performance shortcomings outside the U.S.. However, few details about the OSN are available publicly, and so we set out to reverse engineer Facebook from Web crawls, packet captures, and network measurements [11, 23, 30]. We examine the extent to which network characteristics on paths between users and Facebook can affect performance, and so we locate infrastructure endpoints and measure paths that connect them to users. We also want to understand how often the same data is fetched, and so we capture packets and elicit rules of interaction and analyze information flow.

2.1 Overview of The Facebook Service

Our first step is to describe the relevant details of the Facebook service as a representative example of a commercial OSN, or more broadly, a hosted application that attracts users with a set of features and attracts advertisers, who pay for the privilege of displaying ads targeted to these users. OSNs interconnect users through *friendship* relations and allow for asynchronous communications of user generated content, such as text, photos, videos, or third party OSN application updates over a *social graph*. These communications form the bulk of OSN traffic and the core of OSN functionality.

To simplify the problem and yet accurately represent OSN traffic patterns, we consider three built-in Facebook modes of interaction: *wall posts*, *comments*, and *like* tags. Wall posts or *status updates* allow users to associate text, photos, or videos with their own or their friends' *profiles*. These posts are displayed in chronological order on a user's profile and disseminated to the home page *news feeds* of the user's friends. Comments and like tags (*likes*) are follow-ups to existing posts. These interactions are delivered to the intended users, their friends, as well as previous participants of the interaction chain.

Each user interaction is implemented through one or more connections with Facebook servers. For example, during a wall post a user opens a TCP connection with Facebook, sends the new content, and receives HTML markup to display the post in the browser. Immediately, the intended addressee user is notified of the new post on their wall by a *push* notification through a persistent TCP connection that also supports Facebook chat. Other users discover new posts by periodically *polling* Facebook servers. In either case, a new connection is established to request post HTML markup and an additional connection may be required to download embedded static content such as photos or video.

To support this traffic, Facebook maintains three data centers located in Santa Clara, CA, Palo Alto, CA, and Ashburn, VA [12]. Ashburn servers reduce service latency for East Coast and European users, but the geographic separation poses challenges for Facebook's state consistency. The Ashburn database is a slave to the one in California, and during peak hours, the state of the two can diverge by as much as twenty seconds [22]. To prevent memcached servers from serving stale data, all *read* requests for posts less than twenty seconds old are served from California. The California data centers also handle *write* requests for all new posts. To host static content such as photos and videos, Facebook uses Akamai's Content Distribution Network (CDN).

2.2 Social Graph and Traffic

To understand the performance issues of Facebook's infrastructure, we first characterize the user social graph

Table 1: Characteristics of regional social graphs and history of posts.

Region	Social graph								Interactions per day						
	Users		Node degree						Wall		Comment		Like	Photo	
	Local	Total	To local users			To all users			Num.	Vol.	Num.	Vol.	Num.	Num.	Vol.
	(mil.)	(mil.)	md.	mn.	s.d.	md.	mn.	s.d.		(MB)		(MB)			(GB)
Russia	0.22	0.26	2	5.4	12.3	2	21.6	71.8	6996	1.62	5495	0.54	176	1328	173.2
Egypt	1.05	3.86	5	18.4	48.0	5	29.9	94.2	32587	7.45	15200	1.93	126	2004	817.96
Sweden	2.30	8.59	11	35.0	60.1	12	48.1	94.3	153015	37.5	18518	1.36	353	25910	3360.5
NYC	2.12	2.95	8	24.4	47.6	14	71.0	164.5	415655	71.3	21057	1.30	1875	46685	6055.1
LA	1.56	2.25	6	22.3	46.3	9	62.0	158.6	165264	30.9	18481	30.9	1031	20448	2652.2

and its traffic. We analyze the social graphs and user interactions distilled from publicly available Facebook crawls of user profiles in a number of regional networks, selected for their size, geographic location, and availability of crawls. Table 1 shows network characteristics germane to our problem collected over 30 days starting on 1-Jun-2009, using techniques of Wilson *et al.* [30].

The left half of the Table 1 shows the characteristics of the social graph. The social graph of a regional network is composed of links between users in that network, or local users, and all their friends, those in and out of the regional network. Column two and three show the number of local users and all users in the social graph of each regional network. We observe that networks vary by size as well as the ratio of local and non-local users in their social graphs.

Further, we characterize node degree of local users to other local users and to all users in the social graph in columns four through nine. Because the number of friends varies substantially between users, we characterize the distribution of node degree by median (md.), mean (mn.), and standard deviation (s.d.).

The small difference between in-network and total node degree in the social graphs of the first three networks suggests that their users are mostly friends with other users in the network. In comparison, users in the two U.S. city networks have a higher percentage of out-of-network, or *remote*, friends, which when compared against the high percentage of in-network, or *local*, users in the social graphs of these networks, suggest high relatedness between remote user sets. This observation will help us interpret results in Section 5.

Also helpful in interpretation of results will be a summary of the number and volume of posts made by users in each region in columns 10 through 16. The number of posts and volume of their content are averaged over 24 hours. We observe that in each network, wall posts are the most popular, followed by comments, photo posts, and like tags. Photo posts generate the most upload network traffic, because of the size of embedded content.

2.3 Transaction Analysis

To understand network load, we need to understand how posts relate to network packet traffic. Tho this end, we collect `pcap` network traces of packets exchanged in each type of user post between our lab and Facebook servers. The network traffic signature of each transaction varies with network conditions, so we extract HTTP requests for each transaction. For example, Figure 1 shows a photo wall post (image plus text) as a series of HTTP transfers between the user and symbolic infrastructure endpoints. The number of bytes in transfers marked with a '*' signifies that transfer size depends on the actual size of the photo and length of text being uploaded.

```
TRANSFER  FROM   TO    BYTES     COMMENT
 1        user   CA    1510      #upload script request
 2        CA     user  5703      #upload script
 3        user   CA    136802*   #photo and post text
 4        CA     user  7168      #display markup
 5        user   CDN   495       #display image request
 6        CDN    user  3819*     #display jpeg
```

Figure 1: Transfers of a photo post.

We have also captured packets of push and poll transactions and each type of interaction they deliver. A user's browser periodically polls Facebook at approximately 55 second intervals. Replies to a poll request returns whatever posts have been made since the last poll. The data required to display these interactions is concatenated to make the transfers more efficient, though embedded static content requires separate communications with the CDN.

In summary, we make three observations. First, display markup and upload scripts (for static content) can significantly inflate network traffic beyond the post content volume in Table 1. Second, while the network traffic involved in each post is small on its own, the number of posts, multiplied by their delivery to multiple friends, can result in substantial server load, an effect we investigate in Section 5.2.1. Finally, many interactions, especially those that include static content or concatenate multiple posts, may be comprised of a large number of network round trips that can add up to substantial delays depending on network characteristics.

2.4 Internet Path Analysis

To understand the behavior of Facebook's traffic in the network, we need to understand the characteristics of the Internet paths between users and the OSN infrastructure. First, we discover the locations of endpoints that service interactions in different regions. Second, we measure latency, loss, and capacity on paths between

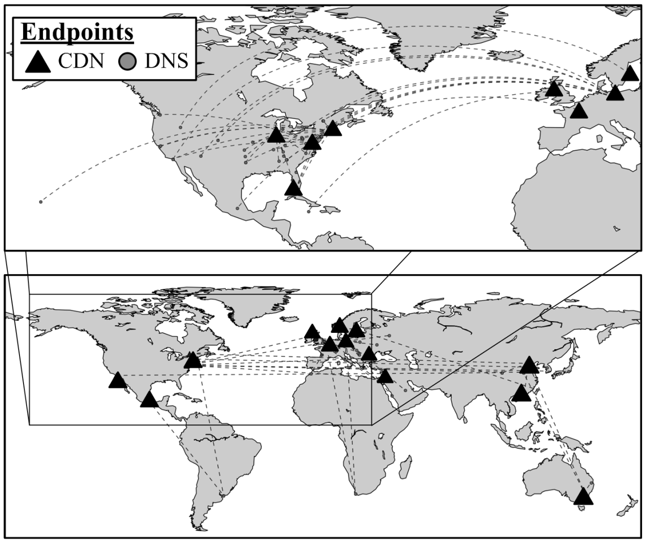


Figure 2: Region to CDN server DNS map.

these endpoints and user hosts in each region. Our goal is to gauge the effect network characteristics can have on OSN traffic.

While the identity and location of Facebook U.S. data centers are well known, the CDN server used in each interaction is determined dynamically through DNS redirection. To discover the set of CDN servers used by Facebook requests we run the following experiment. We iterate over a list of globally distributed DNS servers[1], setting each as our `nameserver` and issuing lookups for CDN hosted Facebook objects. Our approach is similar to the one take by Su *et al.* [23], but instead of issuing requests for CDN objects from PlanetLab nodes, we contact geographically distributed DNS servers directly. We record the Akamai IP addresses returned by each DNS query and create a DNS-CDN server mapping. We then query the IP2Location[2] service for the geographic coordinates of DNS and CDN endpoints and plot the set of unique links in Figure 2. While Su *et al.* discover that a large number of Akamai servers can be used to service requests from a single host, we observe that many of the servers handling Facebook objects are co-located geographically. Our conclusion is that the CDN server handling the request may not be the closest one, for example, when requests issued from the U.S. are sometimes redirected to Europe, and so, the network-induced delay between hosts in Russia, or Egypt, for example, and the CDN server located in Western Europe may be significant. We include the impact of user-CDN traffic in our evaluation of the overall duration of Facebook transactions.

Next, we perform measurements of network latency, loss, and bandwidth to characterize network performance on Internet paths between user hosts and the OSN and CDN endpoints. The performance of "last-mile" access networks can vary between regions and even within regions. To adequately represent this variation, we measure Internet paths to IP hosts placed in each region though IP block country assignments[3] and through the IP2Location service. However, the number of users in each region is larger than the number of host paths we can reasonably measure, and so we characterize the connectivity within each region as a distribution based on last mile path measurements. We found that measurements to around 500 random IP hosts in each region stabilize the shape of that distribution.

Because we do not have access to the actual infrastructure endpoints or the user hosts themselves, for our measurements we rely on PlanetLab[4] nodes in close network proximity to Facebook and CDN data centers. Table 2 lists the infrastructure endpoints and the PlanetLab nodes we have selected as substitutes. The latencies separating Facebook infrastructure endpoints and PlanetLab nodes, shown in the last column, are negligible, and the PlanetLab nodes, which reside in university networks, have high capacity Internet connections. While our approach ignores queuing delays at OSN servers, these are the same for all users regardless of their geographic location and represent a separate problem [18]. For these reasons, we believe that these substitutions do not significantly distort measurements performed from these nodes to user hosts. From each PlanetLab node, we measure latency and packet loss to user hosts within each country using a large number of ping requests.

Accurate available bandwidth and network capacity measurements without the control of both path endpoints is a difficult problem [17]. We make use of an existing measurement effort that estimates last mile capacity from logs of BitTorrent TCP transfers in different regions [11]. We adopt an assumption made in that study that network capacity to hosts within the same /24 subnets is similar and that the last mile capacity is the path bottleneck. Because individual transfers involved in Facebook interactions are small, our evaluation models only consider network capacity, which governs packet serialization delay. Subsequently, we ignore diurnal variations of available bandwidth and asymmetric provisioning of last mile links that affect to a lesser extent "mouse" transfers that comprise OSN traffic.

Table 3 shows means of network measurements in each region. The *direct* (dir.) connections are between user hosts and Palo Alto, CA. The *proxied* (prox.) connections show two numbers: first for paths between CA and a PlanetLab node in each region, and second between the PlanetLab node and regional hosts.

We make two observations. First, direct paths between Facebook and regional users are characterized by

[1] `http://dnsserverlist.org/`

[2] `http://ip2location.com/`

[3] `http://countryipblocks.net/`

[4] `http://planet-lab.org/`

Table 2: Infrastructure endpoint substitutions.

Region	Endpoint	Location	PlanetLab node	Location	Separating latency
Facebook	CA	Palo Alto, CA	planet1.scs.stanford.edu	Palo Alto, CA	2.984 ms
	VA	Ashburn, VA	planetlab3.cnds.jhu.edu	Baltimore, MD	4.228 ms
Russia	CDN	UK	planetlab2.cs.ucl.ac.uk	London, UK	2.217 ms
Egypt	CDN	UK	planetlab2.cs.ucl.ac.uk	London, UK	2.731 ms
Sweden	CDN	Sweden	planetlab2.sics.se	Stockholm, Sweden	1.183 ms
NYC	CDN	Ashburn, VA	planetlab3.cnds.jhu.edu	Baltimore, MD	4.228 ms
LA	CDN	Palo Alto, CA	planet1.scs.stanford.edu	Palo Alto, CA	2.984 ms

Table 3: Network measurement summary.

	Latency (ms)			Loss (%)			Capacity (Mbps)		
Region	dir.	prox.		dir.	prox.		dir.	prox.	
Russia	148	115	31	6.1	0	1.8	29.6	367	29.6
Egypt	164	176	67	5.8	0	5.8	0.92	736	0.92
Sweden	104	95	14	0.32	0	2.9	9.47	188	9.47
NYC	74	43	33	0.75	0	0.6	9.51	99	9.51
LA	27	9.1	18	0.50	0	0.4	2.02	228	2.02

high latency and high loss rates diminishing with proximity to CA. Second, connections tunneled through a regional PlanetLab node isolate most of the path latency on the link to CA and all of the loss onto the regional link. Based on these observations, we believe that TCP proxies deployed at network locations similar to regional PlanetLab nodes would improve TCP performance on paths between Facebook data centers and regional users, and so reduce user request delay.

2.5 Locality of Interest Analysis

While TCP proxies can reduce delay, the other problem is load. Data can be fetched to the region multiple times wasting ISP bandwidth. Our goal is to characterize locality of interest to understand the extent to which this is a problem.

To characterize user communication patterns within each region, we first address a shortcoming of the crawl process: posts not addressed to users within the crawled region cannot be observed in the user *news feeds* scanned by the crawl. Local-to-local (LL) and remote-to-local (RL) user posts can be observed by scanning news feeds of local users. Local-to-remote (LR) interactions can only be observed on the profiles of remote users, which are not represented in the crawls. To estimate the number of LR interactions, we first calculate for each local user u the *reciprocity factor* for interaction type t as: $r_u^t = \frac{1}{|\mathcal{F}_u|}\sum_{v\in\mathcal{F}_u, v\in\mathcal{L}} \frac{|\mathcal{P}^t_{u,v}|}{|\mathcal{P}^t_{v,u}|}$, where $\mathcal{F}_u$ is the set of friends of u, $\mathcal{L}$ is the set of local users, and $\mathcal{P}^t_{u,v}$ is the set of posts of type t made from u to v. We estimate the number of LR interactions as $r_u^t \sum_{v\in\mathcal{F}_u, v\notin\mathcal{L}} |\mathcal{P}^t_{u,v}|$.

Figure 3 shows the breakdown of interactions in each region by user location, marked as a percentage on the y-axis. We observe that posts made by local users, LL and LR, comprise at least half of all posts in each regional network. If posts by local users are delivered to many users within the region, or there exists a high locality of interest, then handling this traffic within the region could reduce OSN service latency.

Delivery of posts is determined by the social graph. A post is delivered to users who are friends with both the poster and the addressee. Figure 4 shows the delivery ratio of wall posts, where the y-axis shows the number of users within the region that receive each post. LL posts result in the largest read-to-write ratio since the sets of poster's and addressee's friends tend to have high overlap within the region. LR and RL interactions have much lower delivery ratio within the region, because friends of the local poster tend to have different sets of out-of-network friends and because these updates do not include self-posts delivered to all the poster's friends.

The delivery mechanisms for comments and likes are slightly different. After a user makes a comment or a like post, subsequent posts are delivered to that user as notifications irregardless of whether the user is friends with the posting users. This cascade of updates can drastically increase the read-to-write ratio of subsequent posts, and so for clarity, we separate the delivery ratio of these posts in Figure 5. Both LL and RL comments and likes can refer to the same news feed item and so consecutive posts quickly inflate the number of readers for each type of interaction. Note that observed ratio of reads to writes would significantly inflate the number of transactions over the writes shown in Table 1.

The cascade of local reads for LR comments or likes depends on the number of RR posts to the same remote user. The RR interactions are not observed by the crawls of local users, and indeed, collecting their set for every remote user might require the knowledge of the entire Facebook state. Subsequently, the number of comment and like updates delivered to local users (for the LR category only) is under-represented in our analysis, and so conclusions we offer later pertain only to the subset of traffic we were able to observe. However, we expect the read-to-write ratio for LR interactions to be comparable to the LL ratio under the assumption that the number of comments made to an LL news feed item is comparable to the number of follow-ups on an RR item. The exclusion of some comment deliveries from our evaluation results in under-estimation of local traffic, but does not substantively impact our conclusions.

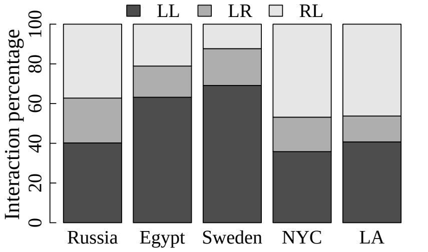


Figure 3: Wall posts by endpoint location.

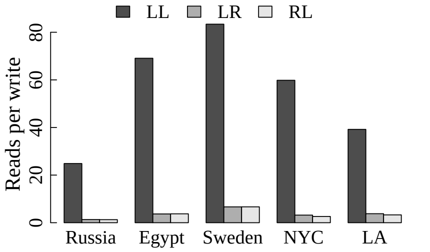


Figure 4: Delivery ratio of wall posts.

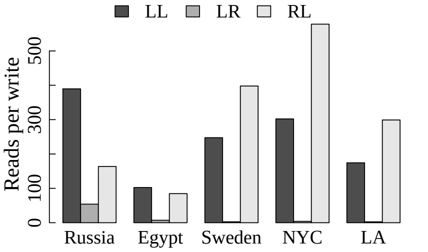


Figure 5: Delivery ratio of comment and like posts.

The analysis of regional interaction patterns shows that traffic local to a region is produced and consumed in significant volumes within the same region. Such a high locality of interest suggests that caching of posts originating in the region would reduce request delay. We believe this shows promise for a redesign of Facebook's infrastructure to cater to local interactions and, by doing so, improve user perceived responsiveness.

2.6 Discussion

Based on reverse engineering Facebook from crawls, packet captures, and network measurement, we have identified causes of lag and wastage of Internet bandwidth. The lag experienced by users interacting with Facebook from outside the U.S. is created when many round trips required in Facebook's communication protocol traverse high latency and loss Internet paths. Further, the high locality of interest means that users in the same geographic region read the same updates created in the region itself. The Facebook infrastructure is wasteful of Internet bandwidth as it delivers the same content, not counting static content delivered by Akamai, many times to a geographic region.

We believe these shortcomings are fundamental to Facebook's design approach, which values consistency of service over performance, and so relies on centralized data centers. The observed shortcomings can be addressed at their core by straightforward and well-understood techniques: TCP proxies and caching. In the next section we offer a proposal to incorporate these techniques in a distributed architecture. At this point, we are interested in quantifying the performance gain that could come from such more distributed architectures and leave optimizations to future work. In Section 5 we will show that performance gains can be achieved without endangering service consistency.

It is germane to relate OSN traffic characteristics to those of other distributed communication applications, such as chat, or email. Unlike chat or email traffic, OSN communications are routinely addressed to hundreds of users, and so OSN traffic is more multicast in nature. Unlike chat, but similarly to email, not all of the addressees may be online when an OSN post is made, and so archival of old messages is required. Unlike email, but similarly to chat, OSN user addresses do not rely on DNS resolution, and so message forwarding is done within the service infrastructure. Finally, similarly to chat and email, OSN users have an expectation of timely delivery of their communications. The combination of multicast traffic patterns, need for persistent storage, timeliness of communications, and lack of a distributed addressing scheme differentiate OSNs enough that it is not clear if such services would benefit from distributed architectures in the same way chat and email systems do [8].

3. DISTRIBUTED FACEBOOK STATE

Through measurement and analysis, we have quantified characteristics of Facebook's service that suggest a distributed infrastructure would improve its responsiveness and efficiency. We now describe two alternative architectures. Both alternatives are based on the deployment of regional servers, but with different functionality. We describe the current Facebook architecture in Figure 6 and the alternatives in the following sections.

The example in Figure 6 shows the geographic locations of endpoints involved in processing Facebook requests of users in Russia. Referring back to Figure 1, which lists the transfers of a photo post transaction, the path labels in Figure 6 specify transfers that traverse these paths. Transfers `1` through `4` traverse the high latency, high loss path between users and U.S. data centers and do not take advantage of either TCP proxies, or the locality of interest.

3.1 TCP Proxies

The long latency and high loss on paths between users and the OSN's servers, and the fact the these path properties can be isolated by regional servers, suggest that TCP proxies would reduce transfer delays [4, 24]. Because the round-trip time between a user host to the regional server (REG) is shorter than to U.S. data centers, the TCP congestion window can expand more quickly and reduce transfer time, especially for the short transfers prevalent in Facebook traffic. At the same time, the TCP connection on the other leg of the path can be shared between transfers, and so use a congestion window expanded in accordance with available link bandwidth. Finally, when losses do occur on a split TCP con-

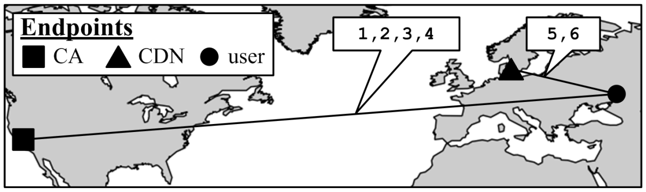


Figure 6: Photo post transfers in Facebook.

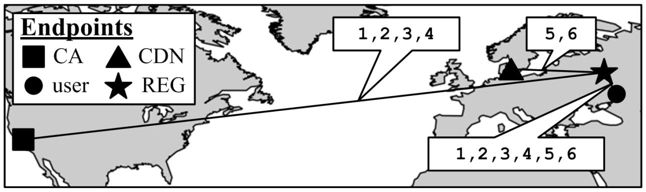


Figure 7: Photo post transfers with TCP proxy.

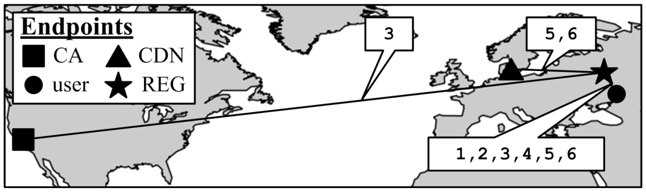
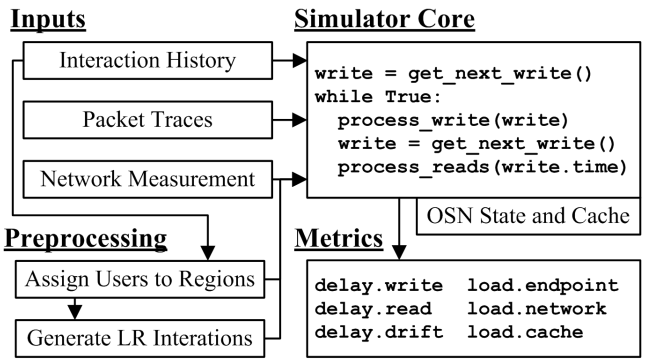


Figure 8: Photo post transfers with OSN cache.



```
write = get_next_write()
while True:
  process_write(write)
  write = get_next_write()
  process_reads(write.time)
```

```
delay.write  load.endpoint
delay.read   load.network
delay.drift  load.cache
```

Figure 9: OSN simulation schematic.

nection, packets are retransmitted only over the shorter sub-paths and incur less delay.

Figure 7 shows the transfer of a photo post forwarded via a TCP proxy regional server. The user host connects to the regional server, which forwards the request to the appropriate infrastructure endpoints. While TCP proxies do not reduce the number of transfers, as compared with Figure 6, these transfers can achieve lower delay over sub-paths connected at the regional server. Our approach is different from one-hop routing [23], because we proxy the connection and consider path loss rate as well as latency. The regional server separates high loss from high latency of a path because loss is found mainly on the last mile links within the region.

3.2 Regional OSN Cache

The high locality of interest of users in each observed region suggests the use of OSN caches at the regional level. Caches allow post delivery to avoid the high latency to U.S. data centers and to reduce network load on these paths. We propose to extend the functionality of regional servers to cache posts and the social graph required for their delivery within a region. Because storing these updates indefinitely would require an amount of storage comparable to that of the U.S. data centers, we time limit the regional cache to the user poll interval of 55 seconds.

Figure 8 shows a photo post processed through a regional cache. As in the TCP proxy solution, the user host communicates with the regional server. However, the upload scripts and HTML markup can be returned by the cache server without contacting U.S. data centers, which requires only the content of Transfer 3 to be sent to the U.S. to maintain global OSN state consistency. Additionally, many read transactions could be served directly form the regional server due to the high locality of interest. This change would eliminate many requests to U.S. data centers altogether. Reductions in long range traffic can save money for Internet Service Providers (ISPs) that can now keep more traffic within their own networks and allow Facebook to negotiate lower Internet access costs to the regional servers. Thus, OSN caches have the potential to translate into a long-term service cost reduction in spite of the initial investment in regional infrastructure.

We note that a practical deployment of regional caches would also require the implementation of ad placement. Regional cache servers send page display markup, and so, embedded ads, which could be placed by running regional algorithms, or in coordination with U.S. infrastructure. In this work, we deal only with automatic delivery of posts through the push and poll mechanisms, which do not carry ad content themselves.

4. EVALUATION METHODOLOGY

To evaluate the degree to which our proposed changes are useful, we quantify the performance of Facebook under three alternatives: the current Facebook architecture, the regional TCP proxy architecture, and the regional OSN cache architecture. To this end, we implement an event-driven simulator that recreates and models OSN traffic over measured Internet paths.

Figure 9 illustrates the simulator components and the stages of information flow between them. As input, our simulator takes user post history, post HTTP traces, and Internet path measurement, described in Section 2. To reconstruct Facebook traffic, the simulator performs two preprocessing steps. First, it assigns users without known location into regions represented in the social graph. We begin with a distribution of users in the known networks. We then assign users to regions based on a probability equal to the fraction of total users in each region. While this method can misplace some users, the overall distribution of users, locality of

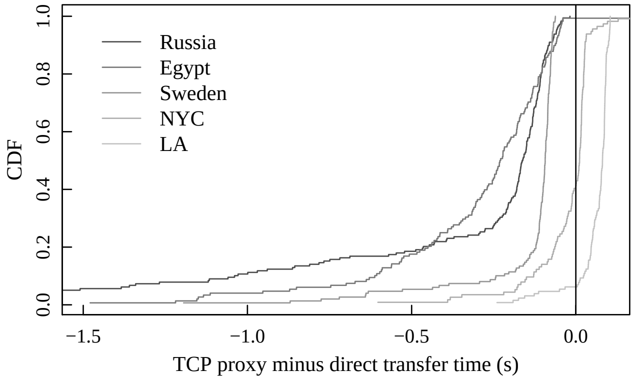


Figure 10: CDF of TCP proxy time saved.

interest, and interaction consumption ratios remain the same. In the second preprocessing step our simulator generates LR interactions through the process described in Section 2.5.

Preprocessing leads to the simulator core, which iterates through the history of user interactions. Depending on the type of interaction, our simulator reconstructs traffic of each post and its delivery through the push or poll mechanisms. The network traffic is replayed over TCP connections modeled on the measured Internet paths to collect a set of delay and load metrics that describe OSN performance. These results become the basis for our conclusions.

In more detail, `process_write()` handles individual write interactions. TCP transfers are simulated over measured paths to calculate interaction delay and set a write's `commit_time` to the California databases. Sorted by `commit_time` processed writes comprise the simulated OSN state. `process_reads()` handles the delivery of writes to active users. We assume that users are active for half an hour at a time [31] and that user activity corresponds with their posts. Writes are delivered as reads only to users active at a write's `commit_time` through push or poll mechanisms. Push message are delivered immediately as a notification to an active user. Otherwise, active users poll every 55 seconds and receive writes that have been committed since the last poll.

4.1 Regional Server Functionality

The differences between the current Facebook architecture and the proposed alternatives are implemented in `process_interactions()`, which simulates the series of TCP transfers required for each interaction in Figures 6 through 8. As illustrated in Figure 7 transfers between an infrastructure endpoint and a user are proxied at the regional server. Data is forwarded on paths between the infrastructure endpoint and the regional server and between the regional server and the user. In calculating the delay of proxied transfers we account for the fact that both TCP connections are established sequentially, but can proceed simultaneously.

Figure 10 shows a CDF of the time differences between proxied and direct 8 KB transfers to Palo Alto, CA from hosts within each studied region. All of the measured hosts in regions separated from the US by transatlantic links, and some hosts in US cities, benefit from tunneling as indicated by the negative difference between transfer times. To avoid penalizing well-connected clients in regions close to Facebook data centers, TCP proxies are used only for hosts that can benefit based on their network connectivity.

To implement the regional cache architecture in our simulator, two extensions are needed. First, we associate `cached_time` with each write to indicate when a post becomes available for push or poll requests. Second, LL and LR writes are sent to REG, cached there on arrival, and forwarded to US for global OSN consistency. The regional cache can service push and poll requests pertaining to LL and LR writes as soon as these are cached. Push and poll requests for RL writes require these posts to be fetched first from US data centers, and so each poll query sent to REG is forwarded to US to check for such posts. Once fetched to the regional cache, `cached_time` of RL writes is set, they are pushed to the addressee, and become available for delivery in subsequent poll queries.

5. RESULTS

We evaluate the degree to which the proposed architecture alternatives improve Facebook responsiveness and reduce resource use. We characterize service responsiveness in terms of interaction delay and OSN state drift, defined as the difference between the time a post is made and the time it is delivered to the addressee. We show that regional servers reduce interaction delay and OSN state drift, which defines user perception of the real-time feel of the system.

We characterize resource use in terms of server and network traffic load, which are tied to hosting costs, and in terms of the cache size requirements of regional servers, which determine their size and initial cost of deployment.

We show that regional servers can lower hosting costs by reducing U.S. data center load and by shuttling data over shorter distances, thereby preserving network resources. We also demonstrate that regional servers require very modest resources, and therefore pose an attractive alternative for infrastructure expansion.

5.1 OSN Service Responsiveness

We characterize service responsiveness in terms of interaction delay and OSN state drift.

5.1.1 Interaction Delay

Figure 11 shows the mean delay of local write interactions (LL and LR) within studied regions. The y-axis

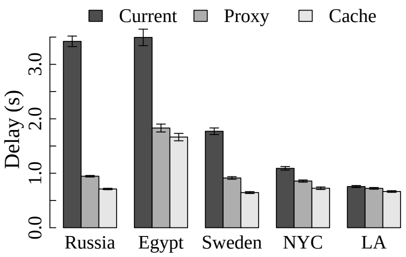


Figure 11: Mean write delay.

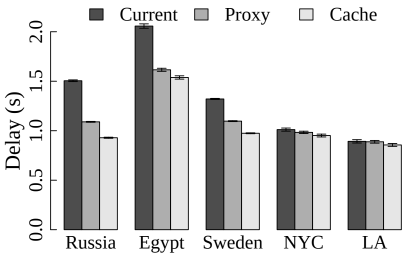


Figure 12: Mean read delay.

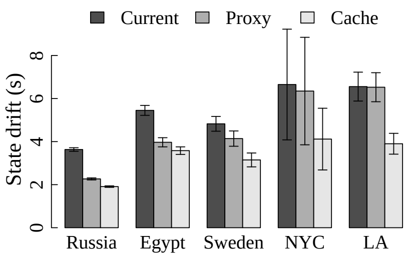


Figure 13: Mean state drift.

marks the mean write delay and 97% confidence intervals separated on the x-axis by region and architecture. Generally, mean write delay decreases in Figure 11 from left to right for regions closer to CA data centers. We observe that for all regions, the current architecture has the highest interaction delay. This result is due to the long latencies and high loss rates on paths between user hosts and U.S. data centers and poor TCP performance on these paths.

Write delay in the TCP proxy architecture is shown in the middle column of each region. Proxied TCP connections reduce write delay by almost 75% in Russia and to a lesser degree in regions located closer to CA as predicted in Figure 10.

The third column in each region shows the write delay when regional caches are present. Caches reduce mean write delay because users communicate with the regional server over low latency links and do not wait for post data to propagate to CA. We observe that the effect of the cache solution as compared to the tunnel solution is most pronounced in Sweden, which has the highest percentage of writes originating in the region as shown in Figure 3.

Figure 12 shows the mean read delay of push and poll requests. The y-axis marks interaction delay separated on the x-axis by region and architecture. Similar, to write delay, read delay decreases the closer the region is to U.S. data centers. While Russia and Egypt are similarly far from U.S. data centers in terms of network latency, the delay of read requests in Egypt is higher. The reason for this difference is that both Russia and Egypt connect to CDN servers in Western Europe, which are relatively farther away for users in Egypt, and so read requests from Egypt take proportionally longer to complete.

As was the case for write interactions, read delay is also shortened by the use of tunneled TCP connections, as seen by the difference between the first and second column for each region. The delay reduction due to TCP proxies is less pronounced for reads than for writes. Write transactions may contain static content such as photos or videos that take a long time to upload to CA. Reads on the other hand, deliver much smaller versions of the same content, compressed for display, and are served from CDN servers, which are much closer to users than CA. As a result, the tunneling and cache mechanism are more effective at reducing mean delay for the larger write transactions that take place over longer and more lossy paths

Finally, the third column of each region in Figure 12 shows read delay when regional OSN caches are used. While caches can eliminate transfers between regional servers and U.S. datacenters, these transfers take place over high data rate paths, and so the reduction to read delay of these TCP transfers is moderate.

5.1.2 OSN State Drift

Read and write delay reflect the responsiveness of the OSN service to user prompts. Another measure of service responsiveness is the time it takes for information to propagate between users. We measure this propagation delay as OSN state drift defined as the difference between the time of the post and the time that the post is delivered to interested users. The reduction in state drift is important because it improves the interactive feel between users and avoids "Can you see it yet?" situations.

Figure 13 shows the mean OSN state drift for each studied region. The y-axis marks state drift in seconds separated on the x-axis by region and the traffic handling method. In general, we notice that OSN state drift is higher for NYC and LA than for the transatlantic regions. This result may seem counterintuitive, since NYC and LA are much closer to Facebook data centers and indeed read delay is lowest for these two regions. However, state drift is affected not only by interaction delay, but in the case of poll transactions, also by poll interval. The high node degree of the social graph in the U.S. and the high node degree of the most active users means that proportionally more interactions in NYC and LA are delivered through the poll mechanism, which inflates state drift.

Comparing the first and second columns in each region, we observe that TCP proxies reduce drift more effectively in regions that are farther away from U.S. data centers. We also observe that serving read requests from a local cache offers a further reduction in OSN state drift, especially in NYC and LA. In these regions, poll requests dominate read delay, and so only the first poll incurs the delay of fetching post content and subsequent polls simply access the cache.

Overall, tunneled TCP connections and regional OSN caches are effective techniques for improving the responsiveness of an OSN service. Tunneling reduces the impact of long network latencies and high loss rates of last-mile links. Caching offers additional delay reductions over tunneling and is especially effective at reducing state drift. Taken together, these techniques have the potential to improve the real-time feel of an OSN by reducing interaction delay in some regions from a noticeable three seconds to a barely perceivable fraction of a second. Because post, push, and poll interactions represent core OSN functionality, we expect our results to translate to all OSN traffic that rely on these mechanisms.

It is important to notice that the consistency of the Facebook service is not violated. First, posts traverse the regional servers to the U.S., and so the the deployment of regional servers does not change the state in U.S. datacenters. Second, user interactions outside the U.S. experience on average less lag and are delivered sooner. Poor performance is no longer the norm, but the exception – we believe that does not constitute a lapse in consistency of the service.

5.2 Resource Utilization

Facebook has demonstrated interest in reducing hosting costs, which are in turn tied to resource use. We characterize resource use in terms of server and network traffic load, and in terms of cache size requirements of regional servers.

5.2.1 Traffic Load

Figure 14 shows the traffic load at each infrastructure endpoint. Each column is a stack of load measurements for each endpoint marked in Mbps on the y-axis. Stacked load measurements are grouped on the x-axis by region and traffic handling method. Because the total volume of traffic is different in each region, we plot the load of each network on its own scale.

In the current architecture, CA data centers handle all the writes, as well as reads, for posts that are less than twenty-seconds-old, which constitutes most of the updates requested by push and poll mechanisms. The CA data center also receives the heavy volume of static content upload traffic. Reads older than twenty seconds are serviced from VA.

The volume of CDN traffic is dwarfed by the previous two categories. Considering the high volume to static content shown in Table 1, this result is unexpected and needs to be interpreted carefully. First, traffic we mark as CDN's includes only downloads, as uploads are sent to CA. Second, the size and resolution of displayed static content can be reduced by as much as a factor of 4 or more from the uploaded image; for example, notice the difference in uploaded and downloaded image size in Figure 1. Finally, our analysis only includes the automatic delivery of information onto users news feed by the push and poll mechanism. User browsing behavior, not included in our traffic, would rely much more heavily on the CDN. In this last point we acknowledge that CDN load in Figure 14 represents only a small fraction of total CDN traffic. Nevertheless our characterization of CA and VA traffic, and CDN traffic for push and poll mechanisms remains valid.

The introduction of the TCP proxies does not reduce traffic to U.S. data centers or the CDN. In fact, the total amount of traffic handled by Facebook increases as the proxy needs to service the incoming and outgoing traffic on the paths from the regional servers to users and to infrastructure endpoints. Notice, however, that proxy traffic on the regional server is less than twice the traffic at other endpoints because lower number of retransmissions on the more reliable paths on either side of the TCP proxy (see Table 3). Finally, we observe relatively lower regional server load for U.S. city networks. The reason for this can be seen in Figure 10, which shows that only a portion of clients in those networks benefit and use proxied TCP connections.

Traffic reduction to U.S. data centers is only achieved with the introduction of regional OSN state caches, which reduce write traffic from regional servers to infrastructure endpoints and satisfy many reads entirely within the regional network. The implication of these observations is that while TCP proxies may increase network costs for Facebook, a partnership between Facebook and regional ISPs could benefit both parties. A similar tradeoff has been used by Akamai, who places their servers for free in ISP networks [23]. Likewise, OSN caches can keep traffic within the region and reduce ISP peering costs. ISPs will pass on at least part of these savings to Facebook, which may want to realize the benefits of regional servers, but may be reluctant to incur extra traffic costs at those servers.

The forwarding of traffic over long distances consumes additional resources that need to be allocated through infrastructure deployment within an ISP or paid for via peering agreements between ISPs. The exact cost of network use depends on network topology, network congestion, and the nature of peering agreements between local and higher tier ISPs.

To simplify the problem, we consider the measure of bytes transmitted multiplied by the distance traveled along the curvature of the Earth, also known as the great circle distance. We calculate great circle distance between server geographic coordinates obtained from the IP2Location service. Although the great circle distance is independent of actual Internet topology, considering the large distances between endpoints it still allows for valid conclusions.

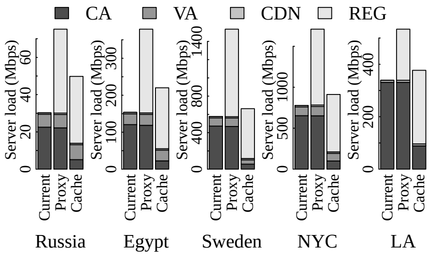


Figure 14: Server endpoint load.

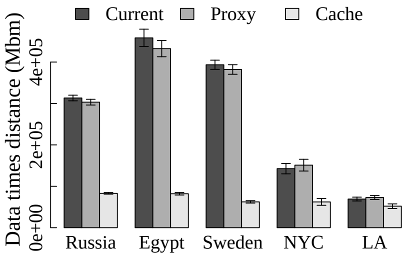


Figure 15: Network load.

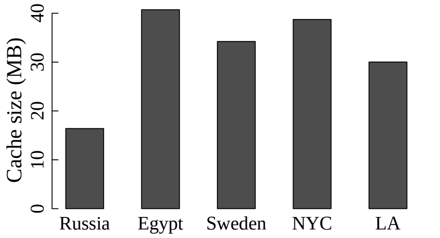


Figure 16: Cache size.

Load multiplied by distance allows us to estimate global impact on network resources consumed by a transfer, as well as the number and size of transfers routed on longer versus shorter paths. The implication is that lower network impact for the same number of interactions makes more efficient use of resources, which should translate to savings for ISPs, and so lower network costs for large customers such as Facebook.

Figure 15 shows the mean network traffic load multiplied by distance in Mbm (Mega-bit-meters) for all interactions. For the European networks the use of TCP proxies slightly decreases data times distance due to fewer retransmissions on the more reliable sub-paths. For the U.S. networks data times distance is slightly higher for the proxied connections, because the effect of fewer retransmissions is lower in the more reliable U.S. last mile networks, and the effect of triangle inequality increases total path distance. The use of regional caches, however, prevents the traversal of the long paths to Facebook data centers and significantly reduces the distance over which data is sent. This effect is most pronounced in regions distant from CA data centers.

5.2.2 *Cache Use*

Improvement in Facebook service responsiveness and reduction in resource use depend on the deployment of regional caches. Besides the already investigated network load of these endpoints, the other component of their cost is the amount of storage required for the cached OSN state.

We calculate cache storage use, or required cache size, as the sum of all the posts stored on the regional cache, where the size of each post is its meta data: ID of poster (8B), ID of addressee (8B), time stamp (8B), target (4B), and item (8B). The target and item fields are used to reconstruct conversation chains. The cache also stores the text of each post, but not the static content such as photos or videos, served by the CDN.

Figure 16 shows the maximum cache storage used at any point during the simulation marked on the y-axis and separated by region on the x-axis. Maximum cache usage depends on the number and size of stored posts within the 55 second poll interval, and so depends on peak load in any such interval. The links of maximum cache usage to peak load makes it hard to correlate cache usage to aggregate regional network properties presented throughout this paper. However, the moderate cache size required to handle peak load in all networks shows that regional caches we propose could be implemented on inexpensive commodity hardware. This result also implies that for the evaluated mechanisms, the major resource requirements are the queries and storage of the social graph. NYC has the largest one at 3.5 GB and could easily be handled by commodity off-the-shelf hardware.

6. RELATED WORK

OSNs consist of social structures and communications, and have been studied from these two perspectives. The structure of the social graph and it's evolution has been studied for a number of the most popular OSNs [2, 3, 13, 15]. At a high level, studies of OSN interactions can be characterized through models of information diffusion [7, 25]. Another approach is to focus on patterns of communications with respect to the social graph by defining a weighted communication graph [30, 9]. Later studies have observed that the communication graph changes over time and so studied its temporal aspects [27, 29]. Finally, recent studies have looked at the interactions of users with OSN services through click-streams collected at ISPs [21, 5].

In our work, we consider network traffic of user interactions within regional networks. Most related are the studies by Nazir *et al.*, who have looked at the network traffic of third-party applications in Facebook, by Liben-Nowell *et al.*, who considered the role geographic proximity of users plays in friendship link formation, and by Carrasco *et al.*, who investigated the role of proximity in email communications [16, 14, 6].

Related to the distributed OSN proposed in this paper is research on optimizing placement data across data centers with respect to user demand [1, 32] and on balancing traffic demands with economic costs of data centers [19, 28]. A recent work by Pujol *et al.* uses social graph structure to more efficiently distribute OSN state within a data center [18]. Our work is different in that we propose distribution based on locality of interest and use short caching time.

Beyond data center based OSN design, a number of projects have introduced interfaces and peer-to-peer designs.[5] Our view is that reliable OSN service is best provided by commercial ventures and single administrative domains, and so we attempt to introduce the benefits of infrastructure distribution to that setting.

7. CONCLUSIONS

The deployment of regional servers to tunnel TCP connections and cache regional OSN state is effective at improving the responsiveness and efficiency of an OSN service. Because of the long network latencies and high loss rates on Internet paths between U.S. data centers and other regions, as well as a high locality of interest in these regions, regional servers are an attractive alternative for the expansion of Facebook's infrastructure. Combining the reduction of traffic to U.S. data centers, the potential to decrease the cost of that traffic, and the very modest storage requirements, cost effective deployment of regional servers could follow a model similar to that used by Akamai, which embeds their servers with ISPs. Finally, judging by the similarity and predictability of performance gains between Egypt, Russia, and Sweden, we believe that the benefits of regional servers will be at least as effective in regions even more distant from U.S. data centers. Further studies might be needed if a precise quantification of results is deemed worthwhile.

8. ACKNOWLEDGEMENTS

We would like to thank Christo Wilson and Krishna Puttaswamy for making available their Facebook crawls.

9. REFERENCES

[1] S. Agarwal et al. Volley: Automated data placement for geo-distributed cloud services. In *Usenix NSDI*, April 2010.

[2] Y.-Y. Ahn et al. Analysis of topological characteristics of huge online social networking services. In *World Wide Web Conference (WWW)*, May 2007.

[3] L. Backstrom et al. Group formation in large social networks: membership, growth, and evolution. In *ACM KDD*, August 2006.

[4] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In *International Conference on Distributed Computing Systems (ICDCS)*, May 1995.

[5] F. Benevenuto et al. Characterizing user behavior in online social networks. In *ACM IMC*, November 2009.

[6] J. Carrasco et al. Agency in social activity interactions: The role of social networks in time and space. *Journal of Economic and Social Geography*, 99(5):562–583, December 2008.

[7] M. Cha et al. Characterizing social cascades in Flickr. In *Sigcomm Workshop on Online Social Networks (WOSN)*, August 2008.

[8] G. Chen et al. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Usenix NSDI*, April 2008.

[9] H. Chun et al. Comparison of online social relations in volume vs interaction: a case study of CyWorld. In *ACM IMC*, October 2008.

[10] Facebook. Statistics. `http://www.facebook.com/press/info.php?statistics`, 2010.

[11] T. Isdal et al. Leveraging BitTorrent for end host measurements. In *Passive and Active Network Measurement (PAM)*, April 2007.

[12] N. Kennedy. Facebook's growing infrastructure spending. `http://www.niallkennedy.com/blog/2009/03/facebook-infrastructure-financing.html`, March 2009.

[13] R. Kumar et al. Structure and evolution of online social networks. In *ACM KDD*, August 2006.

[14] D. Liben-Nowell et al. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–11628, August 2005.

[15] A. Mislove et al. Measurement and analysis of online social networks. In *ACM IMC*, October 2007.

[16] A. Nazir et al. Network level footprints of Facebook applications. In *ACM IMC*, November 2009.

[17] R. Prasad et al. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE Network*, 17(6):27–35, November 2003.

[18] J. M. Pujol et al. The little engine(s) that could: scaling online social networks. In *Sigcomm*, August 2010.

[19] A. Qureshi et al. Cutting the electric bill for internet-scale systems. In *Sigcomm*, August 2009.

[20] D. Schafer. Reducing markup size. `http://www.facebook.com/note.php?note_id=125015758919`, September 2009.

[21] F. Schneider et al. Understanding online social network usage from a network perspective. In *ACM IMC*, November 2009.

[22] J. Sobel. Scaling out. `http://www.facebook.com/note.php?note_id=23844338919`, August 2008.

[23] A. Su et al. Drafting behind Akamai (travelocity-based detouring). *Sigcomm Computer Communications Review*, 36(4):435–446, October 2006.

[24] D. M. Swany and R. Wolski. Data logistics in network computing: The logistical session layer. In *IEEE Network Computing and Applications*, October 2001.

[25] J. Tang et al. Temporal distance metrics for social network analysis. In *Sigcomm Workshop on Online Social Networks (WOSN)*, August 2009.

[26] P. Vajgel. Needle in a haystack: efficient storage of billions of photos. `http://www.facebook.com/note.php?note_id=76191543919`, April 2009.

[27] M. Valafar et al. Beyond friendship graphs: a study of user interactions in Flickr. In *Sigcomm Workshop on Social Networks (WOSN)*, August 2009.

[28] V. Valancius et al. Greening the internet with nano data centers. In *ACM CoNEXT*, December 2009.

[29] B. Viswanath et al. On the evolution of user interaction in Facebook. In *Sigcomm Workshop on Online Social Networks (WOSN)*, August 2009.

[30] C. Wilson et al. User interactions in social networks and their implications. In *European Conference on Computer Systems (EuroSys)*, April 2009.

[31] Z. Yang. Every millisecond counts. `http://www.facebook.com/note.php?note_id=122869103919`, August 2009.

[32] Z. Zhang et al. Optimizing cost and performance in online service provider networks. In *Usenix NSDI*, April 2010.

[5] `http://diso-project.org/`, `http://daisycha.in/`, `http://appleseedproject.org/`