FS2You: Peer-Assisted Semi-Persistent Online Storage at a Large Scale

Ye Sun†, Fangming Liu†, Bo Li*†, Baochun Li‡, Xinyan Zhang§
†*Hong Kong University of Science & Technology*, ‡*University of Toronto*, §*Roxbeam Inc.*

***Abstract*—It has been widely acknowledged that online storage systems within the "cloud" of the Internet provide services of a substantial value to end users who wish to share files of any sizes within a group. Such online storage services are typically provided by dedicated servers, either in content distribution networks (CDNs) or large data centers. Server bandwidth costs, however, are prohibitive in these cases, especially when serving large volumes of files to a large number of users. Though it seems intuitive to take advantage of peer upload bandwidth to mitigate such server bandwidth costs in a complementary fashion, it is not trivial to design and fine-tune important aspects of such peer-assisted online storage in a real-world large-scale deployment.**

This paper presents *FS2You*, a large-scale and real-world online storage system with peer assistance and semi-persistent file availability, in order to dramatically mitigate server bandwidth costs. In this paper, we show a number of challenges involved in such a design objective, our architectural and protocol design in response to these challenges, as well as an extensive measurement study at a large scale to demonstrate the effectiveness of our design, using real-world traces that we have collected. To our knowledge, this paper represents the first attempt to design, implement, and evaluate a new peer-assisted semi-persistent online storage system at a realistic scale. Since the launch of FS2You, it has quickly become one of the most popular online storage systems in mainland China, and a favorite in many online forums across the country.

I. Introduction

The online storage systems fulfill one simple purpose, that is to allow an end user to upload files, of both small or large sizes, to the "cloud" of the Internet, to be shared among a group of interested users. As online storage systems evolve, they have become increasingly popular and intuitive to use. In fact, online storage systems of the current generation are so easy to use that they are instead referred to as one-click hosting services. They generally describe web services that allow Internet users to easily upload files onto the one-click hosts' servers, mostly free of charge. Most such services return a URL that can be shared to others, who can then download the file at a later time. Due to the simplicity and versatility of its user interface, this type of file sharing has rapidly become a favorite among users, overtaking well-known peer-to-peer (P2P) file sharing services, such as BitTorrent.

As online storage systems become increasingly popular, however, server bandwidth costs have become prohibitively expensive, as files are hosted in either content distribution networks or dedicated large data centers. *Rapidshare*, one of the most well-known one-click hosting systems, deployed a total of 1500 terabytes of online storage in its data centers, in Asia alone. Skyrocketing bandwidth costs from server-based architectures have made it necessary for all online storage systems that remain free of charge to impose certain restrictions, including download bandwidth limits per day, file size limitations, as well as maximum file online available time.

Though it may seem intuitive to take advantage of peer bandwidth contributions to mitigate server bandwidth costs, the architectural and protocol design of such a peer-assisted online storage system should not be taken lightly. It is nevertheless non-trivial to design and fine-tune a new system that utilizes peer bandwidth contributions in a complementary fashion, without sacrificing the ease of use, reliability, and performance of one-click hosting. Further, peer bandwidth contributions should be used in a completely transparent manner, while still minimizing server bandwidth costs for serving more popular files. The architectural design should be able to scale to a large number of users, and withstand the test of real-world usage over a long period of time.

In this paper, we present FS2You [1], a real-world online storage system that we have designed, implemented, and deployed to provide one-click hosting services with peer bandwidth assistance. FS2You is designed to dramatically mitigate server bandwidth costs, while maintaining the ease of use and performance comparable to the best server-based solutions. In response to a number of fundamental challenges, we present the architectural and protocol design in our system, and carry out an extensive measurement study to evaluate its performance, based on real-world traces involving millions of users over a long period of time. Since the launch of FS2You, it has quickly become one of most popular online storage systems in mainland China. We describe detailed design elements in this system, and analyze the reasons motivating its performance benefits and popularity. To the best of our knowledge, this paper represents a first attempt in the literature to design, implement, deploy, and evaluate a real-world peer-assisted online storage system, supported by large volumes of measurement traces. We are convinced that this work is of substantial value to obtain an in-depth understanding of how peer bandwidth contributions may be utilized in a complementary manner.

II. Related Work

Although peer-to-peer (P2P) file sharing systems have received significant research attention, there exist a number

*The research was support in part by grants from RGC under the contracts 615608, 616207 and 616406, by a grant from NSFC/RGC under the contract N_HKUST603/07, by a grant from HKUST under the contract RPC06/07.EG27.

of important differences between P2P file sharing and peer-assisted online storage systems. Peer-assisted online storage systems use peer upload bandwidth in a complementary fashion, in order to improve performance and file availability. P2P file sharing systems do not use servers to store actual file content, as all files are exchanged among users. As a result, they have no guarantees on file availability, and files being downloaded may become unavailable at any time when all "seeds" (peers with a complete copy of the file) leave the system. Due to these differences, the design principles and objectives of peer-assisted online storage systems differ substantially from existing BitTorrent-like protocols.

There exists a large number of measurement studies of existing P2P file sharing systems in the literature. Here we briefly discuss three examples. First, Gummadi *et al.* [2] have analyzed a 200-day trace of Kazaa workloads collected at the University of Washington, in order to better understand user characteristics and the file popularity. It is further shown that the popularity distribution of Kazaa files deviates from the Zipf rule, which is caused by the immutability of these files, and the "fetch-at-most-once" user behavior.

Second, Stutzbach *et al.* [3] have studied various aspects of user dynamics by measuring three different P2P file sharing systems: Gnutella, eMule, and BitTorrent. One of the main observations was that, at any point of time, a majority of participating peers are long-lived peers; while the remaining small portion of short-lived peers join and leave the system so rapidly that they constitute a relatively large portion of observed sessions. Third, Guo *et al.* [4] collected traces from popular trackers of BitTorrent to examine peer arrival rates and downloading performance. They showed that file availability in BitTorrent could deteriorate quickly due to the exponentially decreasing peer arrival rate and the lack of "seeds."

Other measurement studies of various aspects of BitTorrent-like P2P file sharing systems (*e.g.,* [5]–[7]) have investigated the tracker availability, file integrity, flash crowd handling, and impact on ISPs. More recently, attention in measurement studies have also focused on YouTube [8], [9], a popular server-based on-demand streaming system, including usage patterns, file characteristics, as well as distribution of requests across videos. Although one can consider YouTube and other similar platforms as online storage solutions specifically designed for multimedia, they are primarily of server-based solutions.

Our work in this paper offers original contributions that are substantially different from all previous works. Granted, a large portion of our work focuses on measurement studies on FS2You. For example, we analyze typical performance metrics and user behavior, and examine the correlation among file popularity, user requests, file sizes, and the efficiency of peer assistance. That said, rather than pure measurement studies that treat real-world systems as "black boxes," our research starts from clear design objectives, followed by proposed solutions that are custom tailored to the specific challenges. As a result, FS2You, a peer-assisted online storage system, has been implemented and deployed in the real world at a large scale, with extensive trace-driven measurement studies. Such a "closed-loop" research methodology has not been previously applied to peer-assisted online storage systems, and is also rarely seen in the literature on server-based online storage and P2P file sharing systems as well.

III. FS2You: Challenges and Design

In this section, we first identify the major challenges as we design FS2You, a peer-assisted semi-persistent online storage system. In response to these challenges, we present the system architecture and main components of our design, including the management of peer topologies (*i.e.,* overlays), the design of peer assistance protocols, as well as server-side strategies.

A. Design Objective and Challenges

Two extremes of the cost-performance tradeoff exist in the design of file sharing systems. P2P file sharing systems provide no guarantees on file availability, while server-based online storage systems are able to make such guarantees, at the prohibitive cost of server bandwidth and storage. The design objective of a peer-assisted semi-persistent online storage system is to achieve a *reasonable and balanced tradeoff* between these extremes, as we conserve valuable bandwidth and storage resources on servers by taking advantage of peer assistance, while still maintaining a semi-persistent nature of file availability, as well as improving the downloading performance. To achieve such an objective, the following challenges need to be addressed:

- ▷ The reduction of server involvement may bring adverse effects on file availability and downloading performance. *How do we substantially conserve server bandwidth costs, while mitigating such adverse effects and maintaining an adequate level of service quality and user experience?*
- ▷ As the system scales up to a large population, we intend to store contents that are as valuable to users as possible, with a limited amount of server storage space. While recognizing that files will be available on a *semi-persistent* basis, *how do we mitigate the drawback of degraded file availability with the limited pool of server storage?*

B. Architecture and Components

Fig. 1 illustrates the basic system architecture and interactions among main components, which include: (1) *The Directory Server*: each file referred as a *channel* is assigned with a unique channel ID. A directory server keeps the information of all channels (files) including the channel ID and the MD5 (Message-Digest algorithm 5, the hash value of the corresponding file); (2) *The Tracking Server*: this maintains the participating peers' information for each channel; (3) *Replication Servers*: FS2You deploys 60 replication servers in China. Replication and content sharing mechanisms will be described later in Sec. III-E; (4) *Peers*: there are two types of peers in the system: (i) peers that upload files to servers, hereafter referred as *uploading peers*; and (ii) peers that download files only, referred to as *downloading peers*. To upload a file, a peer computes the MD5 hash value of the

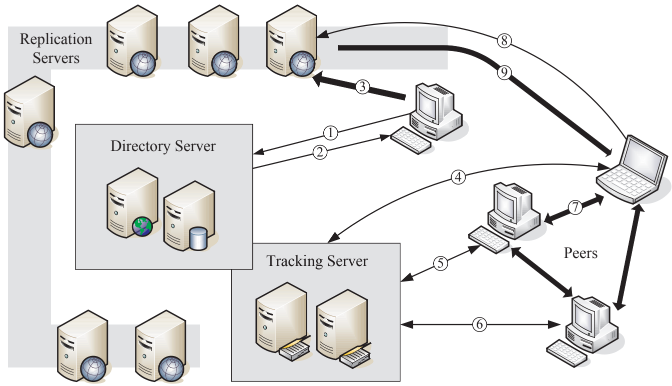

Fig. 1. The hybrid architecture of FS2You system. Arrows 1, 2, and 3 represent the interaction between a peer and servers for uploading new content. Arrows 4, 5, and 6 represent the interaction between peers and tracking server to maintain the overlay. Arrow 7 represents the gossip communication and file blocks sharing among peers. Arrow 8 and 9 represent a peer requesting help from a replication server when necessary.

file and issues request(s) to the directory server. The directory server determines whether the file is new and redirects the file to one of the replication servers. Relevant information is updated in the tracking server and the URL for this file is generated for other peers to retrieve the file.

C. Peer Partnership and Overlay Management

All peers involved in a file, *i.e.*, either downloading the file or holding a replica of the file, are organized into an *overlay* for exchanging block availability information and content sharing. As we will observe in Sec. V, peers in FS2You are highly dynamic and less popular files constitute a large portion of demands. This observation makes it non-trivial to achieve a high level of P2P efficiency. How do we construct and manage the overlay (by judiciously selecting partners for peers), so that peer resources can be utilized as fully as possible?

As our design choice, FS2You combines coarse-grained tracking servers and decentralized gossip methods for constructing and managing overlays. When a downloading peer joins FS2You, it contacts the directory server and tracking server, and obtains a list of 20 randomly selected peers associated with the same channel. These peers become the initial *partners* of the newly arrived peer. This partnership list is periodically updated (every 5 minutes) and new partners can be added. Peer partners can be *active* or *inactive*, which is determined by whether there are actual connections and data block transfers between the peer and its partners.

Peers need to keep a reasonably number of active and inactive partners in order to maintain a sustainable level of downloading efficiency and to be resilient to network dynamics. In FS2You, each peer can have up to 500 partners in its inactive partner pool. In case the size of the inactive partner pool is over 500, a peer will discard *aged* partners or partners that it has failed to establish a connection. The maximum number of active partners per channel that each peer could have is set to 32 in FS2You. Connections to active partners can be broken from time to time due to a number of reasons, such as slow downloading rates and being idle for a long time. Each peer monitors the number of its current active partners. If the number of active partners falls below 16, it triggers the establishment of new connections with inactive partners, thus promoting them to the status of active partners.

How do we maintain accurate lists of peers in each channel on the tracking server? In FS2You, peers report their status to the tracking server every 5 minutes, which contains vital peer information such as a unique peer identifier, its IP address, and information about channels that it has joined. Since each peer can be potentially involved in a large number of channels, to keep overhead low, the reported status only includes information of the top 20 channels, including the channel identifiers and *download ratios*, defined as the amount of file that has been downloaded so far. The top 20 channels represent files that have been downloaded by this peer, which are ranked by a value computed by the combination of the *file size*, *download ratio*, and *the time when the peer joined the channel*. Intuitively, the larger the file size and/or the download ratio is, the higher the ranking is. In addition, channels that the peer joined later gain higher ranking values. Upon receiving status reports from peers, the tracking server periodically updates the corresponding list of peers associated with each channel. Such a periodic refresh of peer lists in each channel (associated to each file) assists peers to gain access to active partners that are most helpful, with a reasonable level of overhead. Consequently, the downloading performance can be improved, and the load on servers can be alleviated.

D. Content Delivery

Each file is divided into fixed size blocks of 256 KB. A *Block Map* (BM) is introduced to specify the availability of blocks at each peer [10]. The periodic exchange of BMs among peers enables them to locate the needed blocks. Each peer can retrieve distinct blocks from active partners simultaneously.

FS2You implements a unique *sequential block scheduling* mechanism as follows:

- ▷ The first block is always fetched from the server. Intuitively, this reduces the latency for obtaining the first block and enhancing the user download experience.
- ▷ Block scheduling is timer-driven with a period of 5 seconds. The selection of this value has to balance between signaling overhead and peer upload bandwidth utilization. A short period incurs extra signaling overhead, while a long period could potentially under-utilize peer upload bandwidth. While FS2You is specifically designed for use in mainland China, where DSL peers rarely enjoy upload capacities exceeding 512 Kbps, the estimated minimum time for transmitting one block is 256 KB / 512 Kbps $= 4$ seconds. Without being overly aggressive, a conservative value of 5 seconds is chosen. In each round of block scheduling, a peer sequentially requests missing blocks up to the number of its current active partners.
- ▷ To improve file availability and the download experience, peers are allowed to request help from replication servers, but only when any of the following three conditions hold, in order to prevent server bandwidth abuse. (1)

There are currently no active partners, *e.g.*, the file is unpopular or a peer fails to establish connections with any of its partners. (2) None of the active partners hold the desired block. (3) The aggregate download rate from active partners (*i.e.*, the size of data that was downloaded from active partners over the previous scheduling period) falls below 10 KB/second. The 10 KB/second threshold is empirically determined to prevent peers from aggressively consuming server bandwidth.

E. Server Strategies

Servers in FS2You not only provide online storage, but also cooperate with content delivery. There are three strategies that servers adopt to facilitate storage and content sharing: uploading, downloading, content replication and replacement.

$\triangleright$ *Uploading service.* In FS2You, users are allowed to upload a variety of files to servers without *any* size or format limitations. This attracts millions of users to upload a huge volume of content to FS2You, catapulting it to one of the most popular online storage systems in China in a short period of time. Our measurements showed that 500 GB to 1 TB of files are routinely uploaded per day. To cope with such a demand without consuming excessive resources, the following two strategies are adopted: (1) When a user requests to upload a file, the system ensure that only one copy is stored in one of the servers; and (2) this copy is stored in the server nearest to the user requesting the upload. This helps to reduce the uploading time, and to mitigate unnecessary cross-AS traffic.

$\triangleright$ *Downloading service.* Servers complement peers to supply file blocks, especially to those peers suffering poor downloading rates, *e.g.*, below 10 KB/second. The challenge, however, is how to properly satisfy the potentially large number of requests without incurring prohibitively high bandwidth costs. In FS2You, when a server receives a block request, it makes its decisions based on the following policies. (1) If the request is for the first block of a file, it will be served immediately. (2) The request for other blocks will be served in a probabilistic fashion, based on the popularity of the file. Specifically, a *file popularity index* is computed for each file periodically, which is inversely proportional to the number of references to this file during the previous period. The rationale behind this is that, a larger number of references will likely result in more copies of the file at different peers, implying that the servers should serve less. This simple policy implicitly allows peers involved in popular channels to largely rely on peer assistance rather than servers, and allocate more server resources to unpopular files with fewer peers. In our forthcoming measurement studies in Sec. V, we will examine how the strategy influences file download rates and user experience with different popularity levels.

$\triangleright$ *Content replication and replacement.* With a limited pool of server storage place, files in FS2You need to be *semi-persistent* in nature. We wish to mitigate the drawback of such a semi-persistent nature of file availability, and to maintain a high level of user satisfaction. The rule of thumb we have followed in FS2You is to maintain the availability of recently or frequently accessed files, while replacing less popular ones when necessary. Files with a reasonable level of user demand remain available as long as they do not impose an overwhelming server load.

In particular, the following strategies are used in FS2You: (1) Small files (with a size below 10 MB) will not be deleted unless specifically demanded by the original uploading user. (2) Each file i is assigned a reference index H_i, which monitors the ratio between uploaded file sizes and file access frequencies. More specifically, let S_i be the size of file i, and F_i be its daily access frequency, *i.e.*, the daily number of unique IP addresses that have accessed file i. The reference index H_i is calculated per day as $H_i = S_i/F_i$. In FS2You, if H_i is lower than a particular threshold h (empirically set to 100), the file is either small or frequently accessed, and as such should remain persistent in the servers. On the other hand, if H_i is higher than the threshold h for a sustained period of time (set to five days in FS2You), the file will be removed from the servers. The rationale is to store large files only if substantial user interests and popularity persist, in order to avoid excessive use of server storage.

IV. FS2You: Collection of Traces

In order to validate the effectiveness our architectural design, we have implemented FS2You and made it readily available for users (mostly in mainland China) to use. It has quickly become one of the most popular online storage services in China after its deployment. To evaluate and analyze the performance of FS2You, we have implemented a detailed instrumentation mechanism, which is presented in this section.

Each peer in FS2You is designed to report its activities and status to the trace server, using the HTTP protocol. The trace server appends the time of receipt to each report, and then store it locally in log files, with a maximum size of 64 MB in each file. Traces on the order of hundreds of Gigabytes are collected every month. For example, 350 GB traces have been collected by the trace server from 3.3 million FS2You peers, over a one-month period from June 21 to July 18, 2008.

In our subsequent measurement studies, we focus on analyzing two types of reports: *Download Event Summary* (henceforth referred to as "*summary*") and *File Source Snapshot* (henceforth referred to as "*snapshot*").

The *summary* records important statistics between the time when a peer opens a channel (*i.e.*, starts downloading), and when the peer closes the channel (*i.e.*, completes or aborts downloading). The *summary* captures the following: (1) the peer and channel IDs; (2) the size of the file being downloaded; (3) the amount of data downloaded so far; (4) the time instants when the peer opens and closes the channel; (5) the time of the download completion; and (6) the amount of data that are directly served by servers, rather than by peers.

The *snapshot* records statistics about files that a peer contributes, and is reported periodically (every hour). The critical information in the *snapshot* contains the reception time of the snapshot at the trace server and the download ratio. As we stated in Sec. III, the download ratio represents the percentage

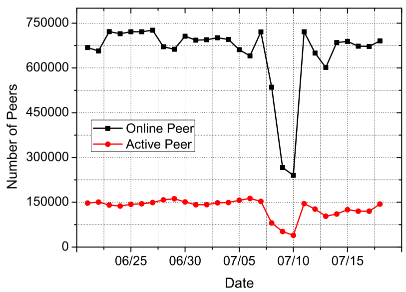

Fig. 2. The number of FS2You online peers and active peers from June 21 to July 18, 2008. The sharp decrease on July 8, 9, and 10 was due to the crash failures of the trace server.

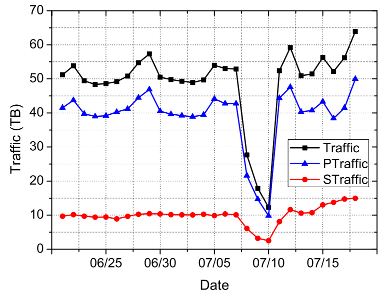

Fig. 3. The total traffic (Traffic), P2P traffic (PTraffic) and server traffic (STraffic) of FS2You from June 21 to July 18, 2008. The sharp decrease on July 8, 9, and 10 is due to the crash failures of the trace server.

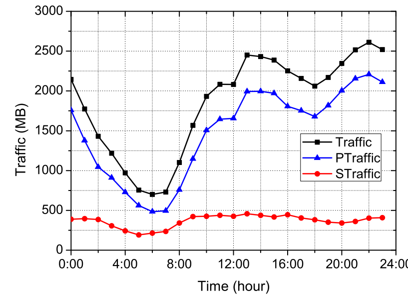

Fig. 4. The evolution of total traffic (Traffic), P2P traffic (PTraffic) and server traffic (STraffic) of FS2You over time on June 21, 2008.

that the file has been downloaded by the reporting peer so far. Tab. I illustrates an example of snapshot, in which a peer with peer ID P_1 reports a snapshot to the trace server at time t_1. At the time of reporting, P_1 locally stores two files with channel IDs F_1 and F_2, and their download ratios are 100% and 50%, respectively. This indicates that P_1 holds a complete replica of F_1 and 50% of the blocks in F_2, both of which can be served to other peers.

TABLE I
SNAPSHOTS: AN EXAMPLE AS REPORTED BY ONE OF THE PEERS.

Time of reporting	Peer ID	Channel ID	Download ratio
t_1	P_1	F_1	100%
		F_2	50%

It is inevitable that there could be inaccuracies in our instrumentation and trace collection mechanism, due to clock skew, crash failures of peers or/and trace server. For example, the trace server indeed crashed and service was interrupted on three days in July. The coarse granularity of reporting snapshots (once per hour) is designed to reduce reporting overhead as the system scales up, but it also introduces a degree of inaccuracy when it comes to estimating the period of time that a peer remains online. We are convinced that the large volume of traces that we have collected is valuable even with such imperfections due to real-world complications, as we shall demonstrate in the next section.

V. Measurement Results and Analysis

To extensively evaluate the architectural and design choices of FS2You, we now take advantage of the large volume of traces we have collected from over three million real-world users, and analyze our measurement results to study a number of important aspects of FS2You, including system dynamics and user behavior, file characteristics, and server involvement. Our measurement studies will close the "loop" of our research methodology, and will validate the effectiveness of our design.

A. Overall Scale and Performance

To demonstrate the system scale and overall performance, we first present some statistics of FS2You online peers, as well as the evolution of traffic.

In FS2You, *online* peers can be classified into two categories based on their activity: *active* peers with download activities, and *inactive* peers that stay online without download activities. For example, if a peer downloads a file on July 1, then we say that the peer is active on July 1; conversely, if a peer stays online on July 1 without issuing any download requests, we say that the peer is inactive on July 1. Among the total 3384948 online peers captured from June 21 to July 18, 2008, 2240517 peers were active for at least one day, whereas the remaining 1144431 peers were inactive during the entire month. Fig. 2 shows the large number of online peers and active peers over the month. The sharp drop on July 8, 9, and 10 is due to the crash failures of the trace server as mentioned in Sec. IV. Interestingly, we found a "weekend pattern" in that the number of online peers decreased remarkably while the number of active peers increased over the weekends. We believe that this is because fewer yet more active peers tend to stay online during weekends.

Fig. 3 shows the amount of traffic over the month, where *STraffic* stands for the traffic served by replication servers, *PTraffic* represents the traffic contributed by peers, and *Traffic* is the sum of STraffic and PTraffic. Again, the sharp decrease is attributed to the missing traces. We have made the following observations. (1) The total traffic of the system varied from 49 TB to 65 TB during the month and also showed a weekend pattern: the total traffic stayed around 49 TB to 55 TB during weekdays and reached its peek around 55 TB to 65 TB during weekends. This is related to a similar observation that there were relatively more active peers with download demand during weekends, thereby leading to higher traffic; (2) Compared to the total traffic, the server traffic was fairly stable and stayed around 10 TB; (3) Over the entire month, up to 80% of the traffic was contributed by P2P delivery, which significantly offloaded the server. A closer look at the daily traffic evolution of FS2You on a representative day is shown in Fig. 4. From 6 a.m. to 1 p.m., there is a steady rise of traffic as an increasing number of users join the system, and the *P2P efficiency* (PTraffic / Traffic) increases from 70% to 85%. Specifically, even during the "cold" period (6 a.m. to 8 a.m.) with fewer users, our design of peer assistance can

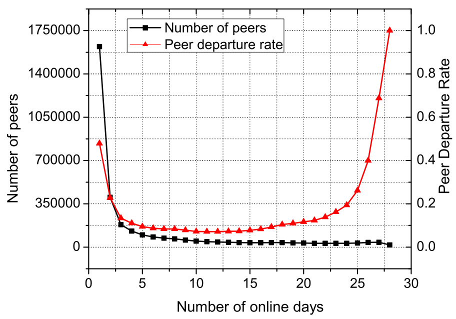

Fig. 5. Peer online day distribution and departure rate from June 21 to July 18, 2008.

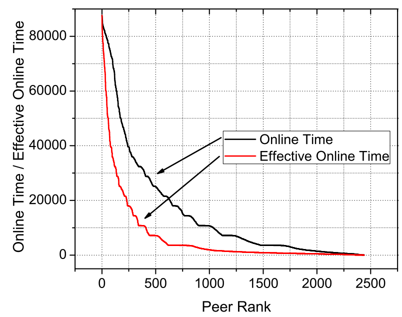

Fig. 6. Peer online time and effective online time distributions (seconds vs. peer rank by online/effective online times) on June 21, 2008.

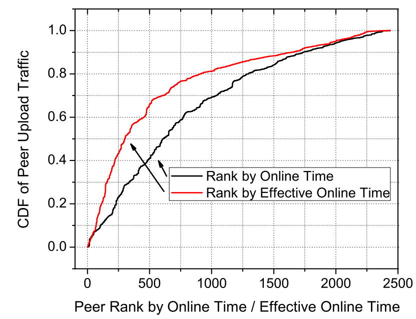

Fig. 7. Cumulative distribution of peer upload traffic vs. the descending ordered ranks of peer online times and effective online times on June 21, 2008.

successfully conserve more than 70% of the server bandwidth cost. For the remainder of the time, the P2P efficiency steadily stayed around 80% and reached its peak of 85.7% at 10 p.m.

In summary, these measurements have testified that our architectural and protocol designs in FS2You can indeed scale to a large number of peers, and to withstand the test of a tremendous volume of traffic (in the order of terabytes per day) over a long period of time. It is evident that the cost of server bandwidth has been substantially saved by peer assistance, one of the important design objectives of FS2You.

B. System Dynamics and User Behavior

To obtain a fine-grained understanding of how peer bandwidth and storage contributions can be utilized under inherent user dynamics, we now characterize the online time and file resource distributions of different categories of peers.

TABLE II
STATISTICS OF ACTIVE AND INACTIVE PEERS.

		Daily		Weekly	
		Number	Per.	Number	Per.
Active Peer	Total	147945		744062	
	Online next day/week	66392	44.8%	298105	40%
	Active next day/week	28284	19.1%	165900	22.3%
Inactive Peer	Total	518354		779614	
	Online next day/week	390523	75.3%	527463	67.6%
	Inactive next day/week	368016	71%	463716	59.4%

First, we observed that over a long period such as one month, active peers consist of a large portion of the entire population. Among the total 3.3 million peers captured in our traces, 66% of them were active. However, within a relatively short period such as a single day, Fig. 2 shows that inactive peers seem to dominate the system. For example, among 666299 peers that had been online on June 21, 78% of them were inactive. The remaining 27 days exhibited a similar phenomenon. It is likely that active peers are highly dynamic while inactive peers are relatively more stable. We show this by analyzing the overlap of peers between two adjacent days and two adjacent weeks, respectively. Table II shows that on average, 75% of inactive peers showed up on the next day and 71% of inactive peers would still be inactive; while only 45% of active peers appeared on the next day and 20% of active peers would still be active. Statistics between adjacent weeks show a similar trend: 70% of inactive peers and 40% of active peers showed up in the next week.

We further capture the system dynamics from another perspective by finding the relationship between peer departure rate (the *peer departure rate* of n days is defined as the ratio of the number of peers who have stayed online for exactly n days to the number of peers who have stayed online for equal to or more than n days) and the number of online days. Fig. 5 depicts the peer online day distribution and peer departure rate for one month. The curve may not be very accurate, since we did not take the online day distribution before and after the measured period into our consideration. Up to 47% of peers stayed for just one day in the system. It also means that after staying online for one day, 47% of peers left the system. This value decreases dramatically as the number of online days increases, and stays around 10% from 10 days to 24 days. It reveals that *young* peers, especially those that newly appear in the system, are more likely to leave the system while *aged* peers (most likely inactive) are relatively stable.

TABLE III
COMPARISON BETWEEN ACTIVE AND INACTIVE PEERS.

Category	Active Peer	Inactive Peer
Online days (day)	4.28	7.63
Online time per day (hour)	5.7	8.7
Resources (measured for 4 weeks)	3.86	1.66
Resources (measured for 1 day)	5.5	2.4
Contribution	20.2%	79.8%

We next compare the online time and file resources between active and inactive peers for a time period as short as a single day or as long as one month. Here we use the maximum number of files recorded in a peer's *snapshot* to represent the amount of resources the peer owns. Table III shows that, on average, an active peer in a day owns 5.5 files while an inactive peer owns only 2.4 files. An active peer stays online for around 5.7 hours in one day, while an inactive peer stays online for more than 8.7 hours. When the period becomes to one month, an active peer stays online for 4.28 days out of 28 days and owns 3.86 files on average, while an inactive peer stays online for 7.63 days and owns only 1.66 files. It shows that inactive peers are more stable but hold less file resources compared to active peers.

In summary, the large population and stability of inactive peers have the potential to be fully utilized in the design. As we closely compare the upload contribution of peers, we found that up to 79.8% of traffic was contributed by inactive ones while 20.2% was contributed by active ones. This demonstrates that the available resources among peers have been well utilized to enhance P2P efficiency and thus alleviate server bandwidth costs.

Finally, to further inspect short-period user behavior and its impact on peer assistance, we focus on a popular channel with 2439 downloading peers on June 21, 2008. Specifically, we are interested in the peer online time distribution and its correlation with peer upload contribution. We use two metrics to capture user behavior after download completion: (1) peer online time (beginning from downloading the file); (2) peer *effective online time* (defined as the duration for downloading the file and lingering with the replica). Fig. 6 compares the distributions of the two types of online times. We can see that many peers tend to have long online times. For example, up to 26% peers stayed for more than 5 hours, and 10% peers even stayed for more than 11 hours. In contrast, the effective online times of peers are relatively shorter, which is demonstrated by the remarkable gap between the two curves. Specifically, the average online time is around 3.6 hours, while the average effective online time is only around 1.4 hours. In addition, we found that as much as 60% of peers had less than half an hour of effective online times. These observations have evidently shown an important user behavior in such a peer-assisted online storage system: many peers *explicitly* removed their downloaded files from their local storage soon after downloading, which makes them, though online, unable to contribute upload bandwidth to the system.

Fig. 7 plots the cumulative distribution of peer upload traffic for this channel versus the descending ordered ranks of peer online times and effective online times, respectively. We can see that the curve corresponding to effective online times is more skewed than that of online times. Specifically, the top 20% of peers with longer effective online times contributed nearly 70% of traffic. This clearly confirms that peer upload contribution is strongly correlated with peer effective online times, rather than peer online times.

C. File Characteristics

We now examine how FS2You files with different characteristics are handled with peer assistance and the server-side strategies, based on the traces from a representative day on June 21, 2008. Among 91530 diverse requests for a variety of files, we found that around 47% of files are compressed archives (*e.g.*, in *rar* or *zip* format), 30% are videos, 12% are audio, and 11% are other types.

First, we apply Zipf analysis to understand the request distribution of FS2You files. Zipf's law states that if objects are ranked by the request count, the popularity of the i-th most popular object is proportional to $i^{-\alpha}$, where α is a constant. Zipf distribution exhibits a linear shape on log-log scale. Fig. 8 plots the file request count versus the descending ordered list of popularity rank on a log-log scale (left y-axis), along with a linear fit curve. We can see that the file request distribution of FS2You does not follow a Zipf distribution. Specifically, the empirical curve is much flatter than the Zipf curve among the most popular files. This implies that the most popular files are significantly less popular than Zipf prediction. We believe this is caused by the immutability of files, and the "fetch-at-most-once" user behavior [2]. Instead, by plotting the empirical data on log-y^c scale (right y-axis), we found that the FS2You file request distribution can be well fitted with a linear line, indicating that it follows the stretched exponential (SE) distribution [11] with a proper constant $c = 0.12$.

Further, we apply concentration analysis [9] that show how skewed the requests from peers are towards popular files. Fig. 9 plots the cumulative distribution of the file request count and the corresponding traffic, versus the descending order of file popularity rank (normalized). We find that the Pareto principle ($80/20$ rule) is applicable with respect to both the requests and traffic, and that the traffic is even more skewed because popular files usually have relative larger sizes, demonstrated in Fig. 11. In addition, we also plot the cumulative distribution of server traffic resulted by these files. We find that for the resulting server traffic this principle does not hold, which implies that peer assistance effectively mitigated the server load accounted for by the popular files.

We next examine the correlation between file popularity and P2P efficiency. Fig. 10 plots the P2P efficiency of each file versus the descending order of file popularity rank (normalized). To facilitate our observation of a global trend, we perform a simple *adjacent-averaging smoothing* process: we divide the entire set of files into a number of groups with similar popularity ranks, and compute the average P2P efficiency for each group of files. As expected, the global trend of the curve shows that more popular files enjoy higher P2P efficiency. Specifically, some highly popular files even enjoy 80% to 90% P2P efficiency, which is very encouraging for the service provider. The increasing noises and fluctuations along the curve imply that the variations in P2P efficiency become larger as popularity decreases. Generally, less popular files have relatively lower P2P efficiency.

We further investigate the correlations of file size, file popularity, and file replicas. Fig. 11 plots the average number of file requests and replicas versus file sizes, grouped into different ranges. We have made the following three observations. *First*, large files (over 300 MB) receive more requests on average than small files (below 300 MB), implying that users' preference in large files. Specifically, 300 MB to 1 GB is the most popular range, which represents typical sizes of videos. *Second*, the server-side replacement strategy (Sec. III-E) effectively guarantees that large files are able to survive in the system only if they are sufficiently popular. On the other hand, to maintain a high level of user satisfaction, a large number of small files can remain available as long

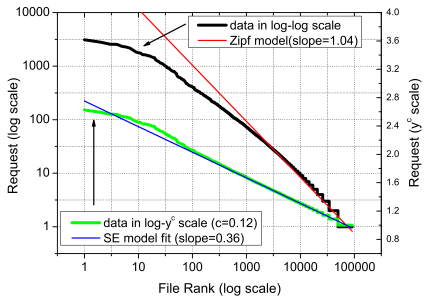

Fig. 8. File request count vs. the descending order of file popularity rank on log-log and log-y^c scale, along with linear fit curves. The empirical data fits stretched exponential distribution, rather than Zipf.

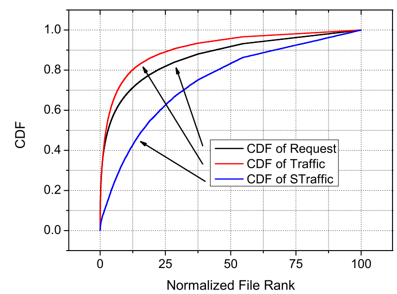

Fig. 9. Cumulative distribution of file request count, the resulting traffic and server traffic vs. the descending order of file popularity rank.

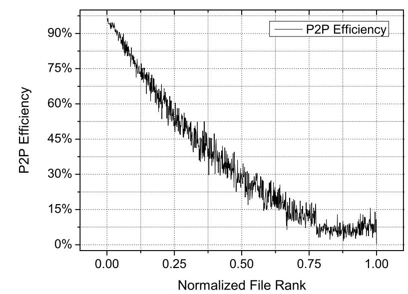

Fig. 10. P2P efficiency vs. the descending order of file popularity rank by applying adjacent-averaging smoothing process.

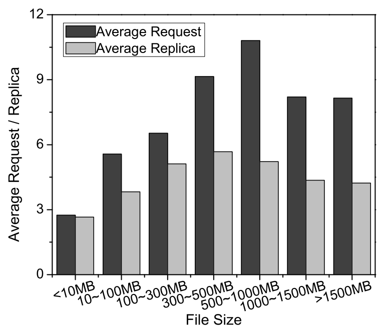

Fig. 11. Average number of file requests and replicas vs. file sizes, grouped into different ranges.

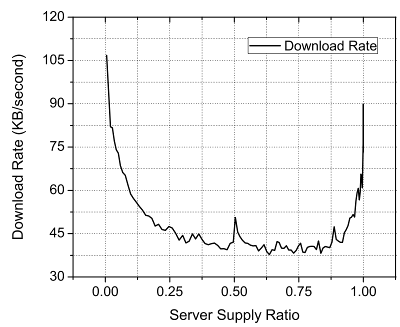

Fig. 12. Average download rate of files as a function of the server supply ratio.

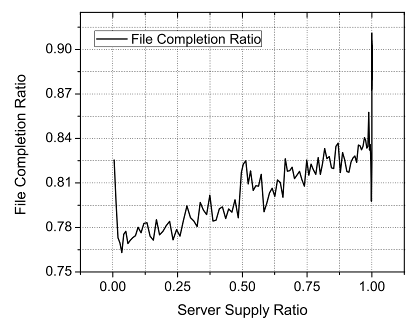

Fig. 13. File completion ratio as a function of the server supply ratio.

as there exists reasonable user demand and they have not occupied excessive storage space. We found that 72.4% of observed files have sizes below 100 MB, and they only occupy 21.4% of the server storage. These results reflect the design philosophy of semi-persistence in an online storage service to balance the trade-off between file availability (thus user satisfaction) and server storage costs. *Finally*, the gap between the number of requests and replicas of large files is larger than that of small files. For example, the average number of requests and replicas are nearly the same for the files with sizes below 10 MB, while the gap increases to a ratio of nearly 2 for the files with sizes between 500 MB to 1 GB. This implies that, although users prefer to download large files, they do not tend to keep such large files in their local storage.

D. Server Involvement Issues

In this subsection, we examine the service quality and user experience provided by FS2you with cost-effective server involvement. Meanwhile, relevant adverse effects are also exposed as the price of mitigating server costs.

We first attempt to emphasize server's impact on file availability, especially for less popular files. Over a one-week observation from June 21 to June 27, we found that around 80% of files are less popular ones (by a *less popular* file, we mean that more than half of the file is supplied by servers). Furthermore, their average request count is 4 while the average request count of all observed files is 14, confirming that they are less popular. These files account for around 25.3% of the total unique requests observed in the one-week trace. In terms of bytes, these files account for around 13% of the total system traffic, and 54% of the total server traffic. More specifically, around 51% of files are completely fetched from servers.

These observations reveal the following: (1) Less popular files represent more than a negligible portion of user demand in FS2You, which reflects one aspect of the inherent nature of online storage services. (2) Due to the lack of partners and replicas, less popular files are usually difficult to be delivered by P2P. To compensate, servers provide better availability for this type of demand with the cost of around 10 TB (61% of the total size of files that appeared in the one-week trace) of storage space. We further extend the scope to all the file requests from peers, and find that up to 95.4% of peers successfully received file blocks; in another word, only 4.6% of peers failed to retrieve file blocks. These again show a superior level of file availability achieved by our design.

Next, we explore the correlation between service quality and the level of server involvement. For each observed file, we define: (1) *server supply ratio* as a ratio of the aggregate traffic supplied by servers to the aggregate download traffic of peers; (2) *average download rate* as a ratio of the aggregate download traffic of peers to the aggregate download time of peers; (3) *file completion ratio* as a ratio of the aggregate download traffic of peers to the product of the file size and the number of requests from peers. Intuitively, the server supply ratio stands for the level of server involvement, the average download rate represents the download performance of peers, and the file completion ratio reflects the satisfaction level of users.

Fig. 12 plots the average download rate of files (KB/second) as a function of server supply ratio. We have discovered the following. (1) Most peers experienced favorable download rates. The average level reaches 66 KB/second and even the lowest rate is above 40 KB/second. (2) Both files that are completely supplied by servers and those that are mainly supported by P2P (with server supply ratio below 0.1) enjoy relatively high average download rates (above 80 KB/second). (3) As we see from the valley of the curve, less popular files with server supply ratios between 0.25 to 0.8 suffer from low download rates (around 40 KB/second).

The results above reveal that the collaboration between servers and P2P could potentially bring negative effects to the service quality under the current design. We believed that this is caused by the peer-side request-from-server threshold (Sec. III-D) and the server-side probabilistic serving strategy (Sec. III-E). In particular, for less popular files (with less peers involved), it could be difficult for a peer to achieve high P2P efficiency and download rates from partners. The reason is that even if the download rate is low, as long as the rate is above 10 KB/second (threshold), the peer can not request help from the server restricted by the design. On the other hand, when the download rate is below 10 KB/second, there is no guarantee that the peer's request from the server be fulfilled given the probablistic serving strategy. Hence, less popular files inevitably suffer from subpar download rates in general.

Fig. 13 depicts the file completion ratio as a function of the server supply ratio. We observed that: (1) the file completion ratio rises from a bottom of 0.77 to a peak of 0.91 as the server supply ratio increases in the mass; and (2) there are two remarkable jumps in the curve. One occurs when the server supply ratio reaches 1.0, while the other one occurs when the server supply ratio falls below 0.02. Since file availability can be guaranteed by servers, the major reason for a low file completion ratio is that peers suffer from low download rates and hence give up the download process prematurely. In this case, the decrease of average download rates shown in Fig. 12 leads to the decrease of the file completion ratio, when the server supply ratio goes down from 1.0 to 0.5. The two remarkable jumps in Fig. 13 correspond to the sharp drop and raise of average download rates when the server supply ratio is near 0 and 1. When the server supply ratio goes down from 0.5 to 0, the average download rate rises while the file completion ratio drops, which we believe is caused by the instability of P2P.

From the above analysis, the FS2You design provides excellent file availability and a satisfactory download experience to a large number of users with cost-effective server involvement. However, it also exposes several design flaws. The threshold of 10 KB/second and the server supply strategy are empirically determined in our design, without fine tuning based on real-world experience. Our observations have revealed an important design tradeoff between the peer download performance and the server load in peer-assisted online storage systems. Intuitively, a high threshold can improve peer download rates, but may potentially incur excessive load on the servers. The server-side probabilistic serving strategy helps to reduce such server load, but may sometimes "leave out some peers in the cold," who indeed need help from servers. How to find an optimal strategy to balance both sides in large-scale systems with millions of users is a challenge subject to future research.

VI. Conclusions

The online storage system has rapidly become one of the most prevailing content sharing services over the Internet due to its simplicity and versatility. Such a service is largely offered free of charge, which remains as a major attraction among the Internet users. This, however, incurs excessive bandwidth cost, and consequently results in various service restrictions. It is natural to consider leveraging bandwidth and storage contributions from peers. This paper, for the first time, describes a large scale real world peer-assisted semi-persistent online storage system, FS2You. The fundamental challenge is to take advantage of peer bandwidth contributions and semi-persistent content storage for substantial cost savings, while at the same time maintaining high service availability and downloading performance at a large scale. We present the architecture and protocol design of FS2You, and demonstrate how the challenge is realized with peer assistance and server deployment in a complementary and user transparent manner. The effectiveness of the system design is verified through an extensive measurement study, which further reveals a number of interesting observations on user behavior, file characteristics and server involvement.

References

[1] [Online]. Available: http://www.rayfile.com

[2] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," in *Proc. of ACM Symposium on Operating Systems Principles*, Oct. 2003.

[3] D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," in *Proc. of ACM Internet Measurement Conference (IMC 2006)*, Oct. 2006.

[4] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-like Systems," in *Proc. of ACM Internet Measurement Conference (IMC 2006)*, Oct. 2005.

[5] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The Bittorrent P2P File-sharing System: Measurements and Analysis," in *Proc. of International Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2005.

[6] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution," in *Proc. of ACM Internet Measurement Conference (IMC 2007)*, Oct. 2005.

[7] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher, "Availability in BitTorrent Systems," in *Proc. of IEEE INFOCOM*, May 2007.

[8] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," in *Proc. of ACM Internet Measurement Conference (IMC 2007)*, Oct. 2007.

[9] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube Traffic Characterization: A View From the Edge," in *Proc. of ACM Internet Measurement Conference (IMC 2007)*, Oct. 2007.

[10] B. Li, S. Xie, Y. Qu, Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the New Coolstreaming: Principles, Measurements and Performance Implications," in *Proc. of IEEE INFOCOM*, Apr. 2008.

[11] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang, "The Stretched Exponential Distribution of Internet Media Access Patterns," in *Proc. of ACM Symposium on Principles of Distributed Computing (PODC 2008)*, Aug. 2008.