Appears in *Proceedings of the First Workshop on Real Large Distributed Systems* (WORLDS 2004)

Deployment of a Large-scale Peer-to-Peer Social Network

Mao Yang†*, Hua Chen*†*, Ben Y. Zhao*§*, Yafei Dai*†*, and Zheng Zhang*‡
†*Peking University, Beijing, China*
§*U. C. Santa Barbara, Santa Barbara, CA*
‡*Microsoft Research Asia, Beijing, China*
{*ym,dyf*}*@net.pku.edu.cn, ravenben@cs.ucsb.edu,* {*i-tochen,zzhang*}*@microsoft.com*

Abstract

We present the design and architecture of the Maze file-sharing and social network. Maze is one of the first large-scale deployments of an academic research project, with over 210,000 registered users and more than 10,000 users online at any time, sharing over 140 million files. Maze includes an evolving incentive structure, and simple mechanisms for providing network locality. We outline the Maze architecture and describe initial results from a measurement study.

1 Introduction

The peer-to-peer model was made popular by file-sharing applications such as Napster [7] and Gnutella [2]. Recent file-sharing applications such as Kazaa [9], Overnet [10] and BitTorrent [4] provide improved scalability and performance. Since then, a number of efforts have been made to better understand their operation and the impact of different incentives and mechanisms on performance, scalability and user behavior [12, 13]. However, the highly distributed indexing and querying of these protocols has limited researchers to indirect measurements and inference at edge nodes.

To better understand the operational properties of these systems, we describe the design, implementation and deployment of Maze, a peer-to-peer file-sharing application with support for network locality and evolving incentive policies. Maze is similar in structure to Napster, with a centralized, cluster-based search engine, augmented with a social network of peers. This hybrid architecture offers keyword-based search with simple locality-based download optimizations. Maze relies on a set of incentive policies driven by direct user feedback from public forums such as BBSes. These policies successfully encourage sharing between using, avoiding the free-loading problem plaguing many similar networks. Finally, MAZE also support node authentication and NAT-traversal, both described in more detail in [3].

Maze is designed and engineered by our academic research team. With control over source code and the ability to deploy software updates, we can leverage Maze as a large-scale measurement platform. Additionally, the centralized query processing and metadata indexing gives us access to information on all files stored in the system as well as all query traffic. This allows us to precisely measure user query patterns, file metadata and size distributions, and monitor changes in user behavior following mechanism and policy changes.

Maze is in its 4^{th} major software release, and is currently deployed across a large number of hosts inside China's internal network. As of July 2004, Maze includes a user population of 210,000 users and supports searches on 140 million files (20 million unique) totaling over 226TB of data. At any given time, there are over 10,000 users online, and over 2700 active searches or transfers occurring simultaneously.

The paper is structured as follows. First, we present the motivation and design of the Maze network in Section 2. We then discuss our deployment experiences and some lessons learned in Section 3. Next, we present initial measurement results in Section 4. Finally, we discuss related work in Section 5 and conclude in Section 6.

2 The Maze Network

In this section, we begin by describing the motivation behind the Maze system and its goals. We then describe the design of the Maze system and discuss its incentives structure.

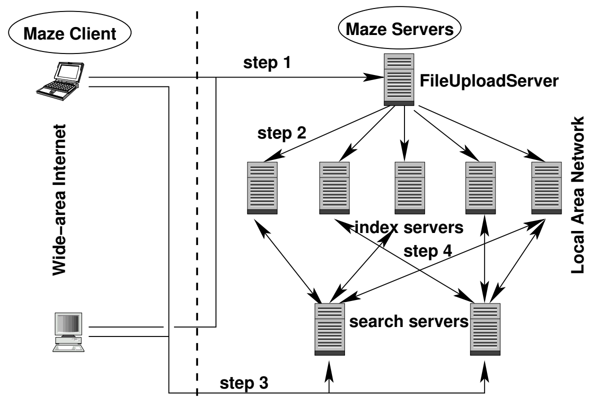

Figure 1: Operations in Maze: 1) Clients upload file metadata to a Maze server; 2) Metadata is replicated to a subset of index servers, where they are indexed; 3) Clients send queries to Maze search servers; and 4) Queries are resolved by index servers.

2.1 Background

FTP servers across the high-bandwidth CERNET [1] provide a large amount of publicly accessible software and documents to educational computing users in China. To address the problem of locating documents across these servers, we built a search engine called T-Net. While it was successful and well-received, T-Net did not solve the basic problems of FTP servers: limited bandwidth and availability.

Feedback from T-Net users led to the design of *Maze*, a peer-to-peer file-sharing application designed with four goals in mind. First, the network should locate replicas in nearby networks whenever possible for efficiency. Next, it needs to reduce the occurrence of "free-riding," where users quickly log off after downloads to minimize their contribution of resources. This requires a strong incentives mechanism that encourages users to share their resources. Third, Maze should leverage social relationships between users to improve efficiency of searches and quality of results. Finally, we want to retain full control over code and deployment so that we can leverage Maze as a platform to experiment with different designs, incentive and security policies, and as a source of detailed file-sharing measurements.

[1] CERNET stands for China Education and Research Network, and is similar to InternetII in structure and bandwidth. Bandwidth between peers ranges from 64kb/s to 2Gb/s. More information can be found at `http://www.cernet.edu.cn/`.

2.2 Maze Design

Default operation under Maze is similar to that of the Napster file-sharing network. In general, each shared file has a small set of associated metadata fields, including owner ID (OID), file name, file type, size, date of creation, and a MD5 [11] hashed signature. Searches on any combination of these fields are possible, including wildcard searches and range queries.

A collection of index servers store information about all files available on peer nodes, regardless of whether a particular node is online at the moment. When clients first come online, they send heartbeat beacons to one of the Maze heartbeat servers, along with an update of which files they have currently available by signature. These are compared to those stored on index servers, and additional metadata is sent for newly acquired files. Metadata is partitioned via simple hashing by OID into a subset of the index servers, where it is indexed by a number of fields for fast searching.

Each search server receives client queries and forwards them to all index servers. It then filters the search results against a list of nodes currently online and replies to the client. The client then contacts multiple replicas to perform a "swarm download," where fragments of the file are downloaded simultaneously from different sources. Querying all indices means that the system returns a positive search match even if only a single replica matches. These steps are shown in Figure 1.

Maze enhances locality of search results by matching the location of replicas with that of the client using IP address similarity. By default, Maze returns those search results first where the replica location's IP address matches the client's address on the first 24 or 16 bits. This equates to preferring hosts in the same class C or class B network, and provides a simple but effective way of localizing file transfers within local area networks.

Maze adds NAT-traversal mechanisms to allow users behind firewalls and NAT boxes to communicate by forwarding through their peers [3]. In addition, users can build a social network of "friend lists" by adding users based on query results or user IDs. Users can browse friends' libraries to find files based on common interests, or forward queries on the social network using a Gnutella-style search algorithm. Whenever a peer accepts a download request for a local file, that request is forwarded to peers on its friends list, and the client could swarm download across all result replicas. Finally, a peer sends keep-alive heartbeats to each peer on its friends list, maintaining connectivity and searching functionality when the central Maze servers become unavailable.

2.3 Incentive Model

File-sharing applications have generally faced the challenge of dealing with "free-riders," users who log on, download their desired files and quickly log off to conserve their resources [1]. To enforce fairness, researchers have examined the issue of incentives from a variety of perspectives, ranging from game-theoretic [8] to practical application-specific [4, 5] approaches.

In Maze, we use an incentive system where users are rewarded points for uploading, and expend points for successful downloads. Our approach is novel in that the algorithm for calculating "points" has evolved over time as a result of direct feedback from the user community. Maze has an extremely active user forum (similar in form to a BBS), where users actively communicate with their peers. The exact parameters of our algorithm were agreed upon by the user community in large online discussions. The exact algorithm is as follows:

1. New users are initialized with 4096 points.
2. Uploads: +1.5 points per MB uploaded
3. Downloads: -1 point per MB downloaded within 100MB, -0.7 per additional MB between 100MB and 400MB, -0.4/MB between 400MB and 800MB, and -0.1 per additional MB over 800.
4. Downloads requests are ordered by: $T = requestTime - 3 * logP$, where P is a user's point total.
5. Users with $P < 512$ have a download bandwidth quota of 300Kb/s.

The incentive system was designed to give downloading preference to users with high scores. These users add to their request time a negative offset whose magnitude grows logarithmically with their score. In contrast, a bandwidth quota is applied to downloads of users with lower scores (< 512). Additionally, while we encouraged uploads and deducted points for downloads, we recognized that the majority of bytes exchanged on Maze were large multimedia files, and made the download point adjustment graduated to weigh less heavily on extremely large files. We note that this is consistent with our observation that a large number of users have access to high-bandwidth links (we estimate roughly 75% of users are inside CERNET). While bandwidth bottlenecks between CERNET and the external network limit connections to near-dialup rates (5KB/s), the availability of highly sought-after files in Maze attracts and keeps external users.

Finally, we note that the online user community was a key contributor to the success of our incentive model. Instead of fostering an environment that encouraged users to work individually to cheat the system, the virtual community encouraged active cooperation between users, and associated a level of "prestige" with high point values. Users who shared large libraries of files were "respected" and even revered for their efforts. Clients posting requests for files on the forum were often inundated with responses by users who wanted to upload their files to obtain additional points. User vigilance also prevented others from hacking the software to artificially inflate their point totals, as was done in Kazaa [2].

Date	Registered	Online	Active
09/01/2003 *			
10/06/2003	100	< 10	
11/26/2003	600	50	
01/05/2004 *	1000	100	
03/01/2004	30,000	1600	250
03/20/2004 *	45,000	3400	400
04/20/2004	60,000	5000	1000
05/20/2004	80,000	6000	1500
06/04/2004	100,000	7000	2000
06/18/2004 *	120,000	10,000	2700

Table 1: The growth of the Maze online user population. * denotes the release of a major software revision.

3 Deployment Experiences

In this section, we describe some of our experiences and lessons from deploying and running the Maze system. We released the Maze client software in September 2003 [3]. Since then, Maze has gone through 3 major revisions in less than a year, and gained over 100K registered users. Our current data show over 200K registered users.

3.1 Deployment and Upgrades

As research software, Maze was initially used only by the research team. The next group of adopters included FTP site maintainers, who formed the initial stable core of the user base, attracting new users with high bandwidth links and large amounts of content. A BBS forum was then formed, and attracted a large number of participants who became the next group of adopters. Since then, Maze has grown quickly by word of mouth, in part due to the ability to introduce friends and establish social links. Table 3 shows the rapid growth of Maze.

After deployment, the Maze client software was upgraded three times to improve performance and reduce

[2] Kazaa uses *participation level* to limit the maximum range of a user query. Kazaa-lite and other programs allowed users to permanently set their PL to the maximum value of 1000.

[3] The Maze public release is available from `http://maze.pku.edu.cn`, but is currently available only via China's internal network.

resource utilization. Maze clients automatically detect and download upgrades, but only perform the upgrade after querying the user. The first upgrade was necessary to adjust heartbeat rates between outside nodes and nodes behind NAT boxes. The team initially believed a high frequency heartbeat (every 2 seconds) was necessary to maintain persistent connections across the NAT boundary. Later tests showed a 30 second interval between heartbeats was sufficient. The upgrade significantly reduced network traffic to and from hosts behind NAT boxes. A second upgrade fixed a bug in the client that would cause it to flood heartbeats to the Maze servers. Later upgrades focused on minimizing CPU and memory utilization.

Direct feedback from the user community was crucial in identifying and diagnosing software bugs. Despite internal tests before deployment, new bugs were occasionally introduced in software upgrades. In one instance, a complete rewrite of the Maze server software made it impossible for a number of clients to connect. By actively monitoring the user forum and server connections, the team quickly identified and corrected the problem.

3.2 Misuses of the System

Users used a number of ways to improve their points level in the Maze system. First, some users ran multiple instances of Maze on a single machine, and transferred files between them to artificially boost their point scores. Next, 20% of all users try to get around the incentive structure by switching to new identities when the current point level has dropped significantly following downloads. Third, some users modified metadata on their files to spoof new, highly popular files in the hopes of soliciting additional downloads to boost their points. Finally, some users embed popular search strings in the metadata of their files to make them appear more in other users' search results.

3.3 Lessons Learned

Our experience with the centralized design of Maze yielded some surprises. While centralized servers are generally regarded as communication bottlenecks and single points of failure, that was not an issue with Maze. For example, a central set of servers that maintained heartbeats with client nodes scaled well with the user population. A dual-CPU P-III 733MHz machine only uses 3% of its CPU time to process heartbeat messages for 20-30K nodes, using heartbeat frequencies of 2 per minute and small heartbeat packets ($<$ 200 Bytes). Furthermore, Maze servers did not experience any correlated failures that significantly degraded system availability. Instead, the main limitation of the centralized approach was in the processing power required to update file indices. Using a small number of index servers, we can build an index of 10 million files from scratch within an hour. As the number of files passed 150 million, however, indexing took up to 1 day. More files would imply longer indexing times and less up-to-date file indices.

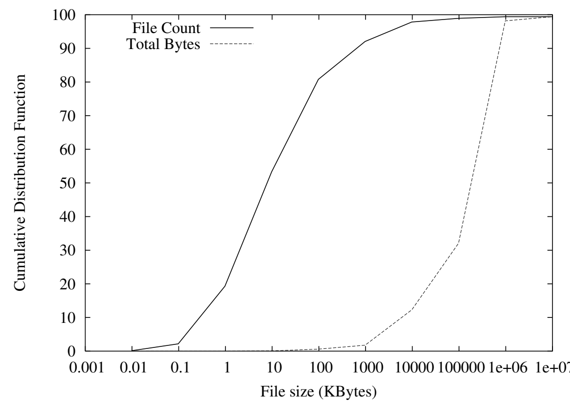

Figure 2: CDF plotted of both the file count and total bytes by file sizes of all files in the Maze network.

Looking back, we can attribute our current deployment success to a number of factors. First, software usability and stability was critical to adoption of Maze by mainstream network users. The user population accelerated its growth when we released our truly stable version in late March of 2004. Second, the virtual community of BBS users provided a way for users to interact on a personal level. It provided a peer to peer support system, and through personal interactions, gave additional weight to the incentive point system. Peers with high points gained social status in the community, and that gave users higher motivation to share content. Finally, Maze's distinguishing features like its novel incentive structure and locality-aware searching attracted users away from existing file-sharing applications.

4 Measurements

Maze currently supports over 210,000 users sharing over 226 TB of data, with the average user sharing over 5 GBs. Because our servers store all file indices and resolve user queries, we have information on every file stored on Maze and can potentially log all requests in the system. Here, we present initial measurement results on the files shared, the structure of the social network, and the number of virtual identities per user.

Ext	# of Files	%TotalBytes	AvgSize(KB)
RMVB	334362	21.88%	158709.68
AVI	275889	21.23%	186626.61
RM	1019203	20.23%	48149.66
MP3	4334838	6.74%	3772.86
ISO	22397	3.95%	427622.73
MPG	135591	2.37%	42428.44
DAT	1156426	2.13%	4475.89
EXE	2398703	2.07%	2091.42
RAR	288717	1.75%	14695.52
WMV	312623	1.58%	12271.59
ASF	126527	1.51%	28987.30
ZIP	626296	1.05%	4062.07
Total	13107386	89.04%	16478.78

Table 2: Files belonging to each type sorted by total size in bytes. Only types accounting for more than 1% of all bytes stored are shown.

Ext	# of Files	%ofAllFiles	AvgSize(KB)
GIF	16038090	11.47%	7.21
HTML	15224072	10.89%	10.11
JPG	9762371	6.98%	141.55
BMP	8290686	5.93%	48.03
M	4555857	3.26%	3.30
MP3	4334838	3.10%	3772.86
DLL	4309781	3.08%	230.00
PDG	4160744	2.98%	30.68
none	3789285	2.71%	223.21
TXT	3184824	2.28%	26.72
WAV	3183987	2.28%	219.07
EXE	2398703	1.72%	2091.42
H	2137047	1.53%	16.24
IRC	1464441	1.05%	3.26
PDF	1331841	0.95%	1214.01
Total	84166567	60.19%	330.77

Table 3: Files belonging to each type sorted by frequency. Only extensions accounting for near 1% of all files are shown.

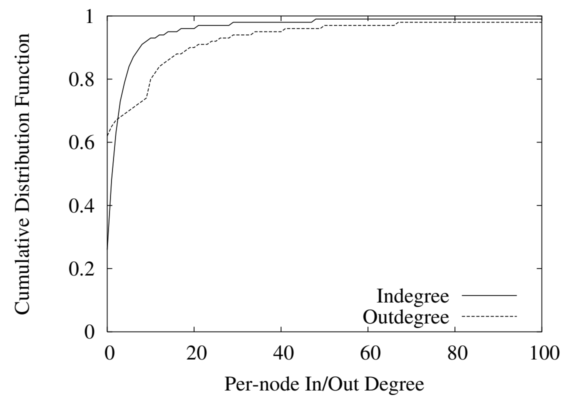

Figure 3: CDF showing the in-degree and out-degree distributions of the Maze social network.

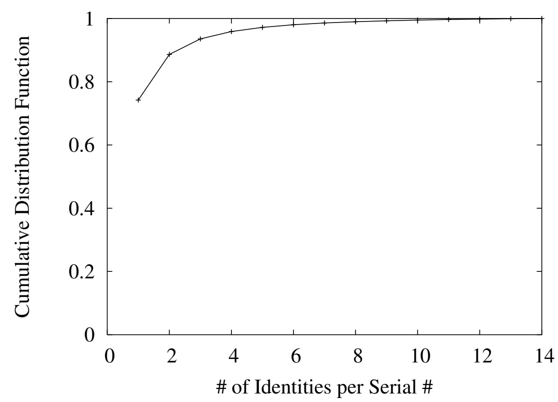

Figure 4: CDF showing the distribution of virtual nodes across machines in the Maze network.

We begin by quantifying the size and type of files being distributed in the Maze network. We plot the distribution of file sizes and count as CDFs in Figure 2. Our results largely confirm those from U. of Washington [12]: a small portion of the total files contribute the large majority of bytes stored.

Unlike previous studies, we can characterize every file stored in Maze according to file type and size. We gather this information from file indices in the Maze index servers, and present them as Tables 4 and 4. Not surprisingly, Table 4 shows the bulk of bytes stored are in the form of large digital movie and music files (AVI, RMVB, RM, and MP3 formats). But Table 4 shows that the large majority of files are small files less than 4MBs. The largest groups of files belong to cached webpages (HTML) and embedded graphics files (GIF, JPG, BMP). In fact, we see that large movie files make up less than 1% of all files (0.24% RMVB, 0.22% WMV, 0.20% AVI).

Next, we examine the structure of the Maze social network. We reconstruct the social network by examining node metadata at the heartbeat servers, and plot the distribution of in-degrees and out-degrees in Figure 3. Outdegree counts the number of nodes in a node's friend list, while a node's in-degree is the number of times it appears in other nodes' friend lists. The snapshot only takes into account social links between live nodes currently in the network. As the data shows, a large portion (60%) of all nodes have empty friend lists while a small portion of Maze nodes have extremely large friend lists. Overall, 66% of all users are connected via social links. We also observe that the social network exhibits properties of a *small-world* network, and has a diameter of 16 hops.

Finally, we wanted to determine how often users run multiple Maze instances on a single physical machine. Users can run multiple virtual nodes on a single machine in order to deceive the incentive system and artificially boost a user's points (see Section 3). Users can also use virtualization to perform a Sybil attack [6] on specific nodes or the central Maze servers.

To quantify virtualization, we need to identify Maze nodes running on the same distinct host. Identification by IP address would be insufficient, since a significant portion of hosts are sharing IP addresses behind NAT boxes. Our solution uses a windows API call that reads and reports the serial number of the local machine's harddrive. This serial number uniquely identifies a machine, regardless of its position in the network.

As shown in Figure 4, the level of node virtualization is relatively low. 74% of all Maze nodes run on unique machines. Other machines run a small number of virtual nodes, enough to locally boost user points, but not enough to perform serious attacks on nodes or servers.

5 Related Work

File-sharing applications first brought attention to peer-to-peer systems. Napster [7] used centralized servers to index available files on its application nodes. Kazaa [9] uses a two-level hierarchical structure, where supernodes stores indices of files shared by nearby clients. Maze differs from these systems in that it supplements the normal network structure with a social network, and uses incentives as a key part of its resource allocation and download scheduling policies.

Mojonation [14] used a virtual currency (mojos) to provide incentive for cooperative sharing. In contrast, Maze uses a community discussion board to actively solicit feedback on the incentive structure from the user population. BitTorrent [4] enforces a modified version of the pair-wise tit-for-tat data sharing model for clients performing simultaneous downloads. However, it only targets clients who are actively downloading the same document.

Maze also differs from existing protocols in using an explicit IP address scoping mechanism to provide locality-aware search results to the end user, resulting in faster downloads and lower bandwidth consumption. Finally, Maze is completely designed, developed, and deployed by an academic research project.

6 Conclusion

Maze is a file-sharing application designed and implemented by an academic research project and currently used by over 210,000 users. In addition to server based file indices, peers connect to each other using a social network, and can rely on friends to resolve queries and forward traffic between hosts behind NAT boxes. Maze uses IP address matching to recognize network locality and encourage downloads from nearby replicas. In addition, Maze leverages an active user forum to determine and encourage the use of an incentive policy.

We are currently working on embedding additional measurement hooks into future client software updates. These updates will give us more insight into client behavior, and will also allow us to use Maze nodes as a distributed measurement platform.

Acknowledgments

The authors would like to thank Yang Zhao and Hanyu Liu for their work on the Maze system, and Professor Xiaoming Li for his guidance.

References

[1] ADAR, E., AND HUBERMAN, B. Free riding on gnutella. *First Monday 5*, 10 (Oct. 2000).

[2] ANONYMOUS. What is gnutella? `http://www.gnutellanews.com/information/what_is_gnutella.shtml`.

[3] CHEN, H., YANG, M., HAN, J., DENG, H., AND LI, X. Maze: a social peer-to-peer network. In *Proc. of CEC'04-East* (Sept. 2004), IEEE.

[4] COHEN, B. Incentives build robustness in bittorrent. In *Proc. of 1st Workshop on Economics of Peer-to-Peer Systems* (June 2003).

[5] COX, L. P., AND NOBLE, B. D. Samsara: Honor among thieves in peer-to-peer storage. In *Proc. of SOSP* (Bolton Landing, NY, Oct. 2003).

[6] DOUCEUR, J. R. The Sybil attack. In *Proc. of IPTPS* (Mar 2002), pp. 251–260.

[7] FANNING, S. Napster. `http://www.napster.com`.

[8] GOLLE, P., LEYTON-BROWN, K., AND MIRONOV, I. Incentives for sharing in peer-to-peer networks. In *Proc. of the 3rd ACM conference on Electronic Commerce* (2001).

[9] KaZaa media desktop. `http://www.kazaa.com`. Using Fasttrack: `http://www.fasttrack.nu`.

[10] Overnet. `http://www.overnet.com`.

[11] ROBSHAW, M. J. B. MD2, MD4, MD5, SHA and other hash functions. Tech. Rep. TR-101, RSA Laboratories, 1995. v. 4.0.

[12] SAROIU, S., GUMMADI, K. P., DUNN, R. J., GRIBBLE, S. D., AND LEVY, H. M. An analysis of internet content delivery systems. In *Proc. of OSDI* (Dec 2002), ACM.

[13] SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking* (2002).

[14] WILCOX-O'HEARN, B. Experiences deploying a large-scale emergent network. In *Proc. of IPTPS* (Mar 2002).