IPTPS '10: 9th International Workshop on Peer-to-Peer Systems

April 27, 2010
San Jose, CA

ENTERPRISE P2P

Summarized by David Choffnes (drchoffnes@eecs.northwestern.edu)

- ***Blindfold: A System to "See No Evil" in Content Discovery***
  Ryan S. Peterson, Bernard Wong, and Emin Gün Sirer, Cornell University and United Networks, L.L.C.

We all rely on easy access to content in the Internet, but a major point of friction in making the content available is that content providers have a responsibility to enforce demands of copyright holders, even if the content was made available by a third party (e.g., users). By making the content easily accessible to users through searches, these providers also expose themselves to potentially costly copyright enforcement. This problem affects a wide range of popular services, from search engines like Google to BitTorrent aggregators and video services such as YouTube.

In this work, Bernard Wong proposed addressing the issue with Blindfold, a system for making providers blind to details of the content that is stored and served while still providing the necessary features of keyword searching that makes content so easily accessible. At a high level, the system divides content discovery into independent index and content services, then uses encryption to hide the search terms and plaintext content. The content discovery protocol further relies on CAPTCHAs to prevent automated recovery of either search terms or the content. In the end, the authors argue that their approach effectively provides plausible deniability for content providers.

The presentation was followed by a lively Q&A session. Predictably, someone asked about legal issues and exactly how far "plausible deniability" goes (e.g., in court). Wong reminded the audience that he is not a lawyer, but the protection could be sufficient for users. The next person wondered whether the system supports anything other than key/value pairs for searching. The answer was no, only key/value for now. The last question returned to legal issues, specifically how to protect index service providers. Wong suggested using cover traffic in the key/value store (i.e., hide real search terms in a sea of fake terms)—for example, using a dictionary to supply them. He also suggested a fully distributed index service, such as the Vuze DHT.

- ***AmazingStore: Available, Low-cost Online Storage Service Using Cloudlets***
  Zhi Yang, Peking University, Beijing; Ben Y. Zhao, University of California, Santa Barbara; Yuanjian Xing, Song Ding, Feng Xiao, and Yafei Dai, Peking University, Beijing

Online storage services are becoming incredibly popular (e.g., through S3 and Mozy), but their reliance on data centers introduces a small number of points of failure that can significantly reduce content availability. Distributed approaches to storage services such as P2P can eliminate this risk of small numbers of points of failure, but the high churn rate of average users in such systems generally reduces availability. Zhi Yang proposed combining these two complementary models to reap the best of both worlds through a system modestly named AmazingStore.

The system is designed using a DHT to locate and store content, with data centers as primary storage and peers as backups. Zhi Yang explained a number of trade-offs in this system, including the level of replication (to ensure a target availability) and the various costs of serving content—free from peers but potentially expensive from data centers. He then discussed how master servers in AmazingStore monitor peer liveness with heartbeat messages to estimate the probability of permanent outages and to replicate data to ensure its availability. Finally, Zhi Yang showed results from a live deployment of their service in China containing 12,000 users and 52,000 objects. The key take-aways were that content availability increased with peer assistance and peers alone cannot provide sufficient availability, so data centers are an integral part of this system.

One important point of confusion about the presentation was whether the system is a peer-assisted data center or a datacenter-assisted P2P storage service. Zhi Yang replied that the system is designed both to improve availability and to offload datacenter costs onto peers. Do users expect the system to provide primarily reliable storage or highly available storage? About 10% of users are storing backups and the remaining portion is for file sharing.

P2P SEARCH

Summarized by Michael Chow (mcc59@cornell.edu)

- ***Estimating Peer Similarity using Distance of Shared Files***
  Yuval Shavitt, Ela Weinsberg, and Udi Weinsberg, Tel-Aviv University, Israel

Udi Weinsberg listed three reasons for not being able to find useful content in estimating peer similarity: not enough metadata, the searches take place in extreme dimensions, and sparseness. The goal, then, is to come up with a new metric for peer similarity.

Their work was done on an active crawl of Gnutella. It looked at a sample of 530,000 distinct songs and 100,000 peers. From the data, they found that 98% of peers share less than 50 songs. They then created a file similarity graph where the files were vertices and the link weights were the number of peers sharing both. The link weights were normalized with popularity and only the top 40% were kept. The filtering process was for filtering out weird tastes. A bipartite graph between every two peers was then created by doing a shortest path walk on the file similarity graph. A maximal weighted matching process was then run on the bipartite graph. It finds the best matching link between files and then normalizes between peers. This serves to eliminate spareness.

Weinsberg then pointed out some issues with this process. He discussed the possibility of the similarity graph not being connected, but this is not really an issue, since this shows dissimilar tastes. Next he talked about the costliness of finding the shortest path on the file similarity graph. In order to reduce the cost of running the shortest path walk, they looked at only the top *N* nearest neighboring files, and they limited the search depth to *K* times the distance of the first finding. They found that 1.5 times the distance of the first finding was sufficient. Weinsberg next briefly discussed the results of their work. They found that there was a correlation between the similarity of artists but no direct correlation between the similarity of geography (physical closeness).

Emin Gün Sirer, Cornell University, pointed out that the voting patterns on Gnutella could be different from sharing patterns and asked how they decoupled this. Weinsberg said that they filtered implicit voting, e.g., user downloaded file, and immediately deleted it, and they also filtered out strange behavior. John R. Douceur, Microsoft Research, asked whether there was a way to make this distributed, because constructing the file similarity graph requires complete knowledge. Weinsberg said it's a problem they are currently working on and would require some sort of approximation of the file similarity graph.

- ***Don't Love Thy Nearest Neighbor***
  Cristian Lumezanu, Georgia Institute of Technology; Dave Levin, Bo Han, Neil Spring, and Bobby Bhattacharjee, University of Maryland

Christian Lumezanu talked about the need for applications to select nodes based on latency constraints. He outlined a scenario involving a multiplayer game where a bunch of players want to find a server that minimizes average latency to players. He described a naive approach which involves polling the latency of servers and exchanging values with each player. The problem with this approach is that not all players may know all of the game servers. He proposed a set of network coordinates that each node has. Latency is measured by the distance between network coordinates. Each node can calculate the theoretical optimum based on the cost function and find the node that minimizes the cost function. The nearest neighbor may not necessarily be enough for complex cost functions.

Lumezanu then introduced Sherpa, an overlay system that finds the lowest cost overlay. Sherpa makes use of Voronoi regions or a set of points in space closer to that node than any other node in that space. Sherpa makes use of compass routing and gradient descent to find the lowest cost. Compass routing is a greedy geometric algorithm that finds the nearest neighbor to the optimum. It selects the node with the lowest geometric angle and stops when the optimal point is in the Voronoi region. Gradient descent is then performed, exploring adjacent Voronoi regions with lower costs. It then checks the nodes in that region. Since there are slight inaccuracies, latency probes are performed for inapproximate mappings to the network space. In evaluating Sherpa, they found that 80% of the time Sherpa selected a cheaper node than the one found by the nearest neighbor. In comparison to an all-knowing oracle, in 65% of the queries, the node chosen by Sherpa is in the lowest 10% of nodes.

Emin Gün Sirer, Cornell, asked if they had looked into ways of minimizing inaccuracies from embedding to the network coordinates. Lumezanu said that these inaccuracies are reduced by using latency probes. John R. Douceur, Microsoft, asked why the nearest neighbor is better than Sherpa 20% of the time. This was mainly due to embedding error.

- ***SplitQuest: Controlled and Exhaustive Search in Peer-to-Peer Networks***
  Pericles Lopes and Ronaldo A. Ferreira, Federal University of Mato Grosso do Sul, Brazil

Ronaldo A. Ferreira addressed the problem of complex queries in P2P networks, which are still an open problem. Existing solutions are based on random walks on unstructured networks, hence incur high traffic on the network due to replication. Moreover, they cannot guarantee that content is found, even if it exists in the network, due to the random nature of the P2P network and lack of complete network coverage. The goal of this work is to have complete coverage of the network when issuing queries by grouping peers and connecting the groups. A query is efficiently propagated among groups until all groups are covered or the answer is found, hence complete coverage is guaranteed.

More specifically, Ferreira suggested grouping peers in such a way that each peer has complete knowledge of the content that exists in each of the other peers in the group (achieved using replications within the group). The groups are placed on a virtual ring, such that each peer in a given group has links to the previous and the next groups in the ring and also to other random groups on the network. When a query arrives in a given peer, the peer checks whether it can be answered with content from its group. Otherwise, the peer sends the query to the connected groups and notifies them which other groups they need to propagate the query to (in case they cannot resolve it), by partitioning the space into non-overlapping intervals.

The size of each group is selected so that it minimizes the index replication and search hops in order to cover the complete set of groups, and was shown to be proportional to the square root of the number of peers. Groups are visited only once, basically performing a broadcast in a randomly constructed tree, determined by the order of groups within the ring. The depth of the tree has a theoretical limit and was shown empirically. Preliminary results show improvement over the previous work BubbleStorm [SigComm 2007].

Overall SplitQuest appears promising—fast, complex queries and less traffic on the network.

Someone pointed out that prior work relies on random search, hence the success rate is smaller than 100%. Why is it that the proposed technique also has less than a perfect success ratio? This is due to peer churn. The results for static scenarios yield a 100% success rate. Someone else said that large networks can result in huge groups. This is a problem both in peer churn, since the groups can change significantly, and in replicating content. Ferreira answered that you don't really share the file but, rather, some metadata about where to find the file in the peers within the same group. You can create smaller groups, but you will have more groups and consequently more messages will be propagated to resolve a query. However, huge groups are indeed a problem and will be considered in future work.

GOSSIPING SYSTEMS

- ***Balancing Gossip Exchanges in Networks with Firewalls***
  João Leitão, INESC-ID/IST; Robbert van Renesse, Cornell University; Luís Rodrigues, INESC-ID/IST
- ***A Middleware for Gossip Protocols***
  Michael Chow and Robbert van Renesse, Cornell University
- ***StAN: Exploiting Shared Interests without Disclosing Them in Gossip-based Publish/Subscribe***
  Miguel Matos, Ana Nunes, Rui Oliveira, and José Pereira, Universidade do Minho, Portugal

No reports are available for this session.

BITTORRENT

- ***Public and Private BitTorrent Communities: A Measurement Study***
  M. Meulpolder, L. D'Acunto, M. Capotă, M. Wojciechowski, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips, Delft University of Technology, The Netherlands

M. Capotă introduced a study on the differences between two fundamental BitTorrent community designs: public (open to anyone) and private (requiring user accounts). Capotă focused on several measurements regarding performance and operational aspects of these communities, namely, download speed, connectability, seeder/leecher ratio, and seeding duration. Not surprisingly, private communities perform better.

The study focused on two public and four private communities, in which the private communities had different sharing enforcement policies. The study was performed using a simplified BitTorrent client that monitored real torrents provided by these different communities. The study showed that private communities have 3 to 5 times faster download speeds; 50% better connectivity (direct reachability of peers in the swarm) on average; at least 10 times higher seeder/leecher ratio at the torrent level; and a much longer average seeding duration. The presenter concluded that tit-for-tat mechanisms to promote collaboration are virtually irrelevant, as private communities appear to be highly successful without requiring such mechanisms.

Was any selfish behavior identified in the study? The authors did not try to identify such behavior, as it was not the main goal of the study. What caused the better connectivity in private communities? This was probably related to more skill on average among the private communities' users. Another question concerned the evolution of swarms for highly seeded torrents. Capotă clarified that this was a subject for future work. Did torrent content affect the study? There was no bias in the study; the studied communities were focused on the same type of content, and thus the comparison is valid.

- ***Comparing BitTorrent Clients in the Wild: The Case of Download Speed***
  Marios Iliofotou, University of California, Riverside; Georgos Siganos, Xiaoyuan Yang, and Pablo Rodriguez, Telefonica Research, Barcelona

Marios Iliofotou reported results on a study to determine if there are more differences between BitTorrent clients than meet the eye. The authors conducted a large-scale study focused on the two most popular BitTorrent clients, uTorrent and Vuze. The study covered 10,000,000 users and 6000 different ISPs during one month last summer.

On average, uTorrent is able to achieve 16% higher download speeds than Vuze, in a consistent way both across time and ISPs. In order to extract some additional clues on the implementation details that could explain this difference, the authors resorted to a controlled setting in order to minimize the impact of hidden variables. The authors were able to identify four main implementation differences, of which two were presented: neighborhood management and upload bandwidth distribution. Concerning the first, Vuze has more ephemeral connection (lasting less than 5 minutes); concerning upload management, uTorrent simultaneously uploads to more peers. Iliofotou concluded with the observation that some design choices have a significant effect on performance and should be carefully evaluated.

Were the authors able to identify traffic shaping in some particular ISPs? This was not the goal of the study, so they did not check for this. Had some clients used more aggressive seeding strategies? Again, this was not an aspect studied in this work. Was peer connectivity taken into account, since different clients could use different tricks to go around firewalls and NAT boxes? They did not check this since their main concern was to avoid other bias sources in the study.

- ***Power-law Revisited: A Large Scale Measurement Study of P2P Content Popularity***
  György Dán, KTH, Royal Institute of Technology, Stockholm; Niklas Carlsson, University of Calgary, Canada

György Dán introduced a study whose goal was to characterize the distribution of content-popularity metrics, in particular instantaneous and the download popularity in BitTorrent systems. To this end the authors conducted a practical study in which they used a Mininova.org screen scrape and a scrape of 721 of BitTorrent trackers.

The study showed that it is hard to fit both popularities in power-law distributions, as previous studies had indicated. Instead, Dán argued in favor of distributions with the following characteristics. For instantaneous popularity, one could perhaps consider a power-law distribution head, a distinct power-law trunk, and possibly (although not obviously) a third power-law distribution tail. For the download popularity, Dán argued in favor of a distribution composed of a flat head, a power-law trunk, and concluded that the tail may exhibit a power law distribution for short periods of time, but the power-law would certainly not hold for long periods.

There were several reactions to the paper. Initially, a participant commented that having more accurate distributions was good, but it was also important to understand their implications in peer-to-peer systems. Dán commented that indeed these distributions were really complex, as they are influenced by several aspects. What changes should be made to peer-to-peer systems given the new, more accurate distributions? This is still an open question, but results show that popularity does not follow power-law distributions, and so systems should clearly not be designed with this in mind. What were the reasons behind the absence of a long tail? This is probably related to content aging, but there were probably other factors. Finally, another participant was curious about the behavior of individual swarm participants. Dán clarified that participants' IDs were not logged in their study, so there was no way for them to know.

- ***Strange Bedfellows: Community Identification in BitTorrent***
  David Choffnes, Jordi Duch, Dean Malmgren, Roger Guiermà, Fabián Bustamante, and Luís A. Nunes Amaral, Northwestern University

Privacy in BitTorrent communities has become an increasing concern. David Choffnes demonstrated this, showing how the strong global connection structure in this P2P system allows an external observer to infer the existence of communities with common interests. This enables a guilt by association attack, where a third party can derive the content that is shared by users of an entire community, using information only from a single member of this community.

In order to evaluate this, Choffnes described a study where the authors took one month of data related to an average of 3000 users per day that established more than 10,000 connections among themselves per day. Using these traces they created a graph where links had weights that were proportional to the number of times they were registered in the trace. Using heuristic-based techniques, the authors were able to identify nine distinct communities with variable sizes. This information allowed the author to infer that information from 1% of nodes had the potential to reveal 80% of the network for direct and one level of indirect observation, and that one host has a potential to reveal 80% of the network for double indirect observation.

Did the authors have any idea concerning the nature of the identified communities? No, but they were looking for explanations for this, such as common geography and content types. To further clarify, they were unable to identify the nature of these communities because the study intentionally did not record the nature of the data being downloaded by users. How could the authors derive the density of the relations between nodes in the same community? This was derived empirically from the data itself. Had the authors tried other techniques, such as clustering, to identify communities? They used simulated annealing to maximize modularity, which is commonly used to identify communities. Did the results show that there was no sharing between members of different communities? Indeed, such sharing activities existed but were not common compared with sharing within communities.