Enabling Efficient Content Location and Retrieval in Peer-to-Peer Systems by Exploiting Locality in Interests

Kunwadee Sripanidkulchai   Bruce Maggs   Hui Zhang
Carnegie Mellon University
{kunwadee,bmm,hzhang}@cs.cmu.edu

Services on the Internet are evolving from centralized client-server architectures to fully distributed architectures. End-hosts are becoming more ubiquitous, more powerful, and more involved in providing services. The wide-spread adoption of Internet access as a utility service is enabling new modes of interaction between end-hosts. End-hosts can provide services as well as use services. We call systems based on such service architectures peer-to-peer systems, and end-hosts participating in such systems peers.

Our interests lie in peer-to-peer content publishing and distribution, where peers publish content to the system and download content from the system. Peers contribute storage and collaborate while participating in the system. Downloading content involves locating peers who have copies of the content, selecting a peer, and retrieving a copy from that peer. The characteristics unique to peer-to-peer systems are dynamicity and variability. For example, content in the system is dynamically replicated, and peers dynamically join and leave the system. Furthermore, peers have a wide range of network access speeds, and variability in load and available bandwidth at each peer can be extensive.

To study variability in performance, we measured ping times to end-hosts on the Internet at 30-second intervals over a 24-hour period. Variability in ping time implies variability in download performance. We collected IP addresses of peers participating in Gnutella [1], a file-sharing application, on April 16, 2001. Out of the 58,400 addresses collected, 2454 were randomly chosen and pinged on April 23 and May 1, 2001. Figure 1 depicts the measured ping time to a peer with cable modem access. The ping times vary over a wide range from 300 milliseconds to 24 seconds. The standard deviation is on the order of seconds, which is typical for a third of the peers measured in our experiments. Unlike servers, end-hosts are not exclusively provisioned for providing service. End-hosts can be used to run many applications locally while actively participating in peer-to-peer content distribution. For many hosts, bandwidth is a scarce resource. Supporting a few concurrent downloads is feasible. But, additional connections can significantly degrade download performance. Protocols designed for peer-to-peer systems need to take into account its dynamic and variable nature.

There are many challenges in designing peer-to-peer content distribution systems. In this work, we address the challenge of locating and retrieving content in a scalable, efficient, and distributed way when peers and the network have extremely high variability in performance. Existing solutions, such as Tapestry [6], Chord [5], CAN [3], and Pastry [4] have addressed scalability. However, no solution explicitly addresses performance. In order to achieve good performance, it is necessary to consider dynamic conditions. Incorporating dynamic performance into existing protocols is not trivial because it can greatly reduce scalability.

We propose a novel solution based on locality in interests to identify a small set of peers for which to maintain dynamic performance state. Peers self-organize into groups. Each peer maintains a list of peers who share similar interests. Peers on the list are ranked based on current interests and dynamic performance. Content is located by querying peers on one's list. Figure 2(a) depicts a peer list overlay constructed on top of Gnutella. When content cannot be found through the list, peers use an underlying location mechanism, such as flooding in Gnutella or lookups in Chord, to locate content. In our initial evaluation, we use the following heuristic to identify locality in interests: peers that have the content we are looking for share the same interests. Figure 2(b) illustrates this relationship. The peer in the middle is looking for content A, B, and C, which can all be found at the peer at the far left.

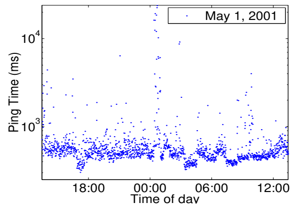
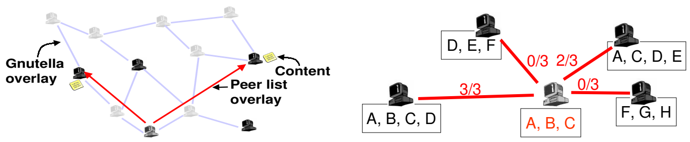
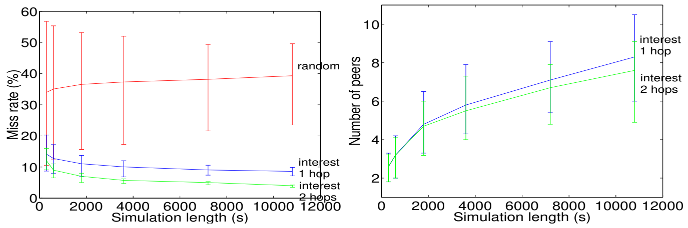


Fig. 1. Ping time to an end-host with cable modem access.



(a) Peer list overlay
(b) Locality in interests relationship

Fig. 2. Using locality in interests.



(a) Miss rate
(b) Peer list size

Fig. 3. Performance of using locality in interests to locate content.

To evaluate the benefits of using locality in interests to locate content, we run simulations using the Boeing corporate web proxy traces [2] to drive the request stream. We compare three content location algorithms: ask random peers, ask peers who share the same interests (1-hop), and ask peers and peers of peers with the same interests (2-hops). The average, maximum, and minimum miss rate observed from 16 simulation using all three algorithms is shown in Figure 3(a). The miss rate is defined as the percentage of requests for which content that already exists in the system cannot be found. Using the random algorithm results in a 35% miss rate. The miss rate using locality in interests is significantly lower: 10% when asking peers 1 hop on the peer list and down to 5% when ask peers 2 hops on the list. Figure 3(b) depicts the average size of the peer list maintained at each node. On average, maintaining a list of 8 peers provides sufficiently low miss rates.

We demonstrate that locating content among peers with shared interests is effective and incurs low overhead. We are currently exploring heuristics to refine our solution by ranking peers in the list based on dynamic performance and boostrapping the list using alternative mechanisms. We are also implementing our solution for Gnutella. Please visit our webpage, http://www.cs.cmu.edu/~kunwadee/research/p2p, for more information about our research and for the implementation we plan to release shortly.

REFERENCES

[1] Gnutella. http://gnutella.wego.com.
[2] J. Meadows. Boeing proxy logs. Available at ftp://researchsmp2.cc.vt.edu/pub/boeing/, March 1999.
[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. *Proceedings of ACM SIGCOMM*, August 2001.
[4] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Submitted for publication*.
[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. *Proceedings of ACM SIGCOMM*, August 2001.
[6] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for wide-area fault-tolerant location and routing. *U. C. Berkeley Technical Report UCB/CSD-01-1141*, April 2001.