A Caching Architecture for Peer-to-Peer Systems

Vana Kalogeraki
University of California, Riverside
vana@cs.ucr.edu, http://www.cs.ucr.edu/~vana
Aditya Mohan
Brown University
a@cs.brown.edu, http://www.cs.brown.edu/~a

As computers become more pervasive and communication technologies advance, a new generation of peer-to-peer (P2P) systems that provide persistence storage and allow users to interact and share distributed resources (*e.g.,* files, CPU cycles) in wide-area environments are increasingly being deployed. Replication and caching are two popular techniques [Krishnamurthy] for reducing network latency and allowing users to fetch data from a peer cache rather than a centralized server. Most of these techniques, however (1) exploit the use of dedicated hardware and (2) assume that the servers are always available and change less frequently. The widespread usage of broadband networks in connection with the low-cost commodity hardware and the maturity of the software technologies, may offer alternative ways for building efficient caching architectures in large scale systems. This goal can be achieved by bringing together multiple, independent, and inexpensive nodes that function together using a peer-to-peer protocol.

Caching architectures for peer-to-peer systems have only recently been proposed. The Freenet project [Clarke] creates multiple object copies in the path between the source and the requesting peer, in order to provide anonymity when publishing or retrieving objects from the system. The Squirrel system [Iyer] proposes web caching algorithms for corporate networks, based on a reliable mapping of a given object key to a unique node in the system. Peer-to-peer systems, however, have some additional challenges, since peers vary in their resource capabilities, behavior and physical location. As a result, the distribution, availability and freshness of the objects may vary significantly.

We propose a new caching architecture for a fully decentralized, self-organizing peer-to-peer system. Our architecture exploits the availability of high performance links to networks, the usage of partial indexes and the replication and caching of popular objects at the peers using local incentives. Nodes maintain a small number of connections with their peers. Peers cooperate to provide the functionality of a decentralized peer-to-peer cache by exporting local caches to other peers and servicing requests on their behalf. Such cache cooperation improves hit ratios and reduces access latencies.

Key to our approach is the routing of the queries and the dissemination of the object updates among multiple peers. When a node receives a request, it first looks in its local cache; if a fresh copy of the object is found there, it is returned. Otherwise, the node generates a `query` message and propagates the request to its peers. These are peers that with high probability will have a copy of the object in their caches. When the object is found, the peer generates a `query_reply` message and propagates it back to the requesting peer. Whenever object caching takes place, the peer refreshes its local index and notifies its siblings so that they can change the contents of their own indexes. This allows our system to balance the load across multiple peers and adapt dynamically to changes in the user behavior or processing and networking conditions.

References

[Clarke] I. Clarke, “A Distributed Decentralized Information Storage and Retrieval System,” University of Edinburgh, 1999.

[Krishnamurthy] B. Krishnamurthy and J. Rexford, “Web Protocols and Practice”, *Addison-Wesley*, Boston, 2001.

[Iyer] S. Iyer, A. Rowstron and P. Druschel, “Squirrel: A peer-to-peer web cache,” *Proceedings of the 21th Symposium on Principles of Distributed Computing*, Monterey, CA (July 2002).