PeerDB: Peering into Personal Databases

Beng Chin Ooi[1], Kian-Lee Tan[1], Aoying Zhou[2], Chin Hong Goh[1], Yingguang Li[1],
Chu Yee Liau[1], Bo Ling[2], Wee Siong Ng[1], Yanfeng Shu[1], Xiaoyu Wang[2], Ming Zhang[1],

[1]Dept Computer Science
National University of Singapore

[2]Dept Computer Science & Engineering
Fudan University

In this demonstration, we present a prototype peer-to-peer (P2P) application called PeerDB[2] that provides database capabilities. This system has been developed at the National University of Singapore in collaboration with Fudan University, and is being enhanced with more features and applications. The concept behind PeerDB is similar to the analogy of publishing personal web sites, except that it is now applied to personal databases. Unlike personal web sites which are usually hosted together in a central web server, personal databases are stored in the person's own PC. In addition, it is increasingly common for people to keep their data in common personal DBMS like MySQL, and MSAccess. Therefore, a PeerDB node allows an user to index and publish his/her personal database for other peers to query.

PeerDB builds on and extends BestPeer [1] for DBMS applications. Briefly, BestPeer is a generic P2P system designed to serve as a platform to develop P2P applications easily and efficiently. It has the following features: (1) It employs mobile agents; (2) It shares data at a finer granularity as well as computational power; (3) It can dynamically reconfigure the BestPeer network so that a node is always directly connected to peers that provide the best service; (4) It employs a set of location independent global name lookup (LIGLO) servers to uniquely recognize nodes whose IP addresses may change as a result.

In the PeerDB network, a set of PeerDB nodes communicate or share resources with each other. Each node comprises four components that are loosely integrated: (a) a data management system (we used MySQL in our implementation) that facilitates storage, manipulation and retrieval of the data at the node, and the associated *local* and *export* dictionaries that reflect the meta-data (schema and keywords); (b) a database agent system called DBAgent that provides the environment for mobile agents to operate on; (c) a cache manager for managing remote meta-data and data in secondary storage; and (d) a user-friendly user interface.

PeerDB has several distinguishing features. First, it allows users to query data without knowing the schema of data in other nodes. To address this issue, we adopt an Information Retrieval (IR) based approach. For each relation that is created by the user, meta-data (keywords/descriptions) are maintained for each relation name and attributes. DBAgents are sent out to neighboring peers to look for potential matches and bring the corresponding meta-data back (and cached). By matching keywords from the meta-data of the relations, PeerDB is able to locate relations that are potentially similar to the query relations.

Second, in PeerDB, we adopt a two-phase agent-assisted query processing strategy. In the first phase, relevant matching relations are identified (as described above). Phase two begins after the user has selected the desired relations. In phase two, the queries will be directed to the nodes containing the selected relations, and the answers are finally returned. The two phases are completely assisted by agents. In fact, it is the agents that are sent out to the peers, and it is the agents that interact with the DBMS. Moreover, a query may be rewritten into another form by the DBAgent (e.g., a query on a single relation may be rewritten into a join query involving multiple relations).

Third, PeerDB can reconfigure the network to keep promising peers in close proximity based on some criterion. Currently, PeerDB supports the *MaxCount*, *MinHops* and *TempLoc* reconfiguration policies. *MaxCount* maximizes the number of objects a node can obtain from its directly connected peers. *MinHops*, implicitly exploits collaboration with peers by minimizing the number of Hops. Finally, *TempLoc* is a temporal locality based strategy that favors nodes that have most recently provided answers.

In the current implementation, PeerDB also provides a secure access to a node's computing resources and databases. Nodes can only access data that are sharable. This is enforced by a security policy that restricts applications to user-specified locations established during platform initialization. Communications between nodes have also been provided with 128 bit encryption to protect the sensitive data from being eavesdropped and viewed as they travel through the PeerDB network.

SIGMOD 2003, June 9-12, 2003, San Diego, CA.

1. REFERENCES

[1] W. S. Ng, B. C. Ooi, and K. L. Tan. BestPeer: A self-configurable peer-to-peer system. In *Proceedings of the 18th International Conference on Data Engineering*, page 272, San Jose, CA, April 2002 (Poster Paper).

[2] W. S. Ng, B. C. Ooi, K. L. Tan, and A. Y. Zhou. PeerDB: A P2P-based system for distributed data sharing. In *Proceedings of the 19th International Conference on Data Engineering*, Bangalore, India, March 2003.