

Partial Selection Query in Peer-to-Peer Databases

Farnoush Banaei-Kashani Cyrus Shahabi
University of Southern California
Los Angeles, CA 90089, USA
[banaeika,shahabi]@usc.edu *

Abstract

In this paper, we propose DBSampler, a query execution mechanism to answer “partial selection” queries in peer-to-peer databases. A partial selection query is an arbitrary selection query that is satisfied with a fraction ϵ of the results; a universal operation with applications in database tuning, query optimization and approximate query processing in peer-to-peer databases. DBSampler is based on an epidemic dissemination algorithm. We model the epidemic dissemination as a percolation problem and by rigorous percolation analysis tune DBSampler per-query and on-the-fly to answer partial queries correctly and efficiently. We verify the efficiency of DBSampler in terms of query cost and query time via extensive simulation.

1. Motivating Problem

In this paper, we discuss the problem of answering *partial selection* queries in peer-to-peer databases, where for an arbitrary selection query, given $\epsilon \in [0, 1]$ a fraction ϵ of the results is returned to satisfy the query. A regular selection query is a specific case of partial selection query with $\epsilon = 1$.

Partial selection is a universal operation, applicable for database tuning, query optimization, and approximate and exploratory query processing in peer-to-peer databases. For example, in a hybrid (structured-unstructured) peer-to-peer database such as [9], to optimize the query plan for a regular selection query, individual nodes can use partial selection as a probe query and estimate the size of the main query as n/ϵ , where n is the size of the partial selection query. Accordingly, a node decides between using the index (i.e., DHT-based search in the structured component of

the system) or initiating a sequential scan (i.e., flooding-based search in the unstructured component) to answer the query; the index is only used for queries with low selectivity. In [9], nodes also use partial selection (there, it is called “sampling”) to tune the database. Each node estimates the frequency distribution of its local stored objects using partial selection queries from the database, and only publishes the rare objects (which are most costly to find) into the database index. Most importantly, partial selection query is a useful and practical query model for direct utilization by the database users. Since peer-to-peer databases are intrinsically open and dynamic computing systems, exploratory/approximate querying is the most appropriate querying mode for these databases. Users often issue several back-to-back queries, each time revising and enhancing the query based on cursory and partial observation of the results of the previous query, just to explore the unknown content of the database and narrow down their search for available useful data. Even when users find their desired formulation of the query, an exact result is most often unnecessary and redundant. Partial selection enables approximate querying that eliminates the redundancy of the unnecessary exact queries to achieve efficiency.

2. Other Approaches

In this paper, we focus on the mechanisms for efficient execution of partial selection queries in *unstructured* peer-to-peer databases. Due to the considerable amount of churn and autonomy inherent in unstructured databases, constructing and maintaining distributed index structures (e.g., a DHT [12]) for such databases is inefficient or even impossible. Therefore, with these databases, in analogy with sequential scan in regular databases, selection queries are inevitably executed by dissemination of the query throughout the network of the nodes and *in situ* evaluation of the query at each visited node to retrieve the relevant data. Consequently, efficient execution of the query is reduced to efficient dissemination of the query throughout the network, or so-called efficient *search*.

* This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0238560 (PECASE). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

There are two main proposals for efficient search in unstructured networks: flooding [8] and random walk [1, 10]. With both of these search mechanisms, query is disseminated throughout the network by recursive forwarding from node to node. With flooding each node that receives the query forwards it to all of its neighbors, whereas with random walk query is forwarded to only one (uniformly or non-uniformly) selected random neighbor. None of these approaches can strike a balance between the two metrics of efficiency for search, i.e., the query cost (communication cost) and the query time (response time). Flooding is most efficient in query time but incurs too high of redundant communication to be practical, whereas a random walker is potentially more efficient in query cost but is intolerably slow in scanning the network. In [13], a two-tier hierarchy is proposed where flooding is restricted to the supernodes at the top tier. This solution only alleviates the query cost of flooding and the problem resurfaces as the top tier scales. In [10], using k random walkers in parallel is proposed as a way to balance the query cost and the query time. However, this proposal does not provide any theoretical basis for selecting the value of k for optimal performance.

Previous search mechanisms are not only inefficient, but also inappropriate for executing partial selection queries. As mentioned above, the main benefit of the partial selection query is that it allows trading off accuracy of the result for better efficiency by limiting the scan of the database to a just sufficiently large fraction of the database that satisfies the query. To enable such a trade-off, a search mechanism that executes partial selection queries should allow adjusting the coverage of the database (i.e., the fraction of the nodes, and hence data objects, visited during dissemination) according to the user specified parameter ϵ of each query. With both flooding and random walk, TTL (Time-To-Live) is the control parameter that can be used to limit the coverage of the network. However, it is not clear how one can adjust TTL according to ϵ for sufficient coverage of the network (where the size of the network is unknown). TTL is often set to a fixed value, a value that is selected in an ad hoc fashion based on the average performance of the typical search queries. In this case, TTL must be re-adjusted as the peer-to-peer database evolves. Alternatively, TTL is gradually increased to expand the coverage, each time repeating the query dissemination from the beginning, until sufficient fraction of the database is covered to answer the query. Although in this case we may be able to cover the proper fraction of the database to satisfy the query, due to the redundancy of repeating the query dissemination, query cost can even exceed that of the regular flooding. Finally, another problem specific to flooding is that the granularity of the coverage is too coarse (the number of covered nodes grow exponentially with TTL), rendering fine adjustment of the coverage impossible.

3. Our Approach: *DBSampler*

We propose using epidemic search mechanisms for efficient execution of partial selection queries in unstructured peer-to-peer databases¹. With epidemic dissemination, query forwarding is probabilistic, i.e., a node forwards a query to each neighbor with forwarding probability p (where $0 \leq p \leq 1$). Therefore, a node may forward the query to zero or more neighbors at each time. Such a query forwarding algorithm is obviously more flexible as compared to both flooding and random walk and subsumes these search mechanisms. The *communication graph* of the epidemic dissemination (i.e., the subgraph of the peer-to-peer network covered by the dissemination) is sparse with small values of p . The communication graph grows larger and denser with larger values of p such that with $p = 1$ the epidemic dissemination is equivalent to regular flooding which covers the entire network.

Our epidemic search mechanism is termed *DBSampler*. *DBSampler* implements the classic SIR (Susceptible-Infected-Removed) epidemic model [5]. Our main contribution with *DBSampler* is derivation of a closed-form formula that given a partial selection query, maps the value of the user-controlled knob ϵ to an appropriate value for the forwarding probability p such that the network coverage is sufficient to satisfy the query (for details of this derivation, refer to our extended paper [2]). Leveraging on this derivation, *DBSampler on-the-fly* and *per-query*² tunes p based on ϵ such that the communication graph of the epidemic query dissemination grows just sufficiently large to cover a fraction of the database that satisfies the partial selection query.

Unlike previous search mechanisms, as required for answering partial selection queries *DBSampler* can cover a certain fixed fraction of the network nodes *independent* of the size of the network. In other words, with a particular value for p size of the communication graph is always proportional to the size of the entire network, such that its relative size (i.e., the covered *fraction* of the network) is fixed. Of course, as mentioned above *DBSampler* can control the size of the covered fraction by tuning p . Intuitively, this occurs because unlike flooding and random walk with which query never dies unless it is explicitly terminated (e.g., when TTL expires), with epidemic dissemination query forwarding is probabilistic and with some non-zero probability each replica of the query may naturally die

-
- 1 In the literature, some times *gossip-based* or *rumor-based* spreading techniques are also termed epidemic techniques [3, 6]. Here, we are not referring to such many-to-many communication techniques, but specifically to the techniques that are modelled after disease spreading in social networks.
 - 2 For each specific query, the value of p is tuned to a fixed value and *all* nodes use the same value for p to forward the query. Varying p as a function of the neighborhood characteristics is part of our future work.

at each step. In turn, the dissemination terminates naturally whenever all replicas of a query die. The larger the network, the more it takes for the dissemination to die and, therefore, the communication graph of the dissemination is proportionally larger.

DBSampler is not only appropriate for answering partial selection queries, it is also efficient in that it strikes a balance between the query cost and query time. Since epidemic dissemination is essentially a flooding-based technique, as our empirical analysis shows, its query time is comparable with that of the regular flooding. Nevertheless, due to the phase transition phenomenon associated with the SIR epidemic model, for the common case of the partial selection queries, the query cost of the DBSampler is up to two orders of magnitude less than that of the regular flooding and comparable with that of the random walk. Intuitively, with epidemic dissemination the dense communication graph of the regular flooding, which with numerous loops represents a large amount of redundant and duplicate query forwarding, is reduced to a sparse communication graph. With fewer loops, the sparse graph contains less redundant paths and therefore, causes less duplicate queries, while covering almost the same set of nodes. Hence, epidemic dissemination can be tuned such that the communication overhead of the flooding is effectively eliminated while its reachability and query time is preserved. Moreover, DBSampler is simple to implement, and since it is a randomized mechanism, it is inherently reliable to use with the dynamic peer-to-peer databases.

The process of epidemic disease dissemination has been previously used as a model to design other information dissemination techniques. Particularly, in the networking community, epidemic dissemination is termed probabilistic flooding and is applied for search and routing in ad hoc networks and sensor networks [7]. We distinguish DBSampler from previous work in two ways. First, although epidemic algorithms are simple to implement, due to their randomized and distributed nature they are difficult to analyze theoretically. For the same reason, most of the previous work restrict themselves to empirical study of the performance with results that are subject to inaccuracy and lack of generality. We employ the percolation theory to rigorously tune DBSampler to its best operating point. Second, those of the few theoretical studies of epidemic algorithms adopt simplistic mathematical models [5] that assume a homogenous topology (a fully connected topology) for the underlying network to simplify the analysis. However, recently it is shown that considering the actual topology of the network in the analysis extensively affects the results of the analysis [4]. We perform our analysis of DBSampler assuming an arbitrary random graph as the underlying topology of the peer-to-peer network and specifically derive final results for a power-law random graph, which is the observed topology for some

peer-to-peer networks [11].

We performed an empirical study via simulation to compare the efficiency of DBSampler versus other search mechanisms. In our experiments, we compared DBSampler with the scope-limited flooding (i.e., flooding with limited TTL) and k -random-walkers (with various k). As discussed above, these search mechanisms are not originally appropriate for the execution of partial selection queries and we had to artificially inform them about the coverage required to satisfy each partial selection query. Our results show that even under such artificial conditions, DBSampler still outperforms scope-limited flooding in query cost while maintaining a reasonable query time. Also, to our surprise, DBSampler not only has a much better query time as compared to that of the random-walk but also outperforms a 32-random-walker (the optimal case as suggested in [10]) even in query cost. See [2] for the detailed empirical results.

References

- [1] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman. Search in power-law networks. *Physics Review Letters*, 64(46135), 2001.
- [2] F. Banaei-Kashani and C. Shahabi. Epidemic sampling for search in unstructured peer-to-peer networks. Technical Report 04-828, University of Southern California, December 2004.
- [3] A. Demers, D. Greene, C. Hauser, W. Irish, and J. Larson. Epidemic algorithms for replicated database maintenance. In *Proceedings of PODC*, August 1987.
- [4] A. Ganesh, L. Massouli, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proceedings of INFOCOM*, March 2005.
- [5] H. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, October 2000.
- [6] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of FOCS*, October 2003.
- [7] L. Li, J. Halpern, and Z. Haas. Gossip-based ad hoc routing. In *Proceedings of INFOCOM*, June 2002.
- [8] Limewire.com. Gnutella, 2004. <http://www.limewire.com/>.
- [9] B. Loo, J. Hellerstein, R. Huebsch, S. Shenker, and I. Stoica. Enhancing p2p file-sharing with an internet-scale query processor. In *Proceedings of VLDB*, September 2004.
- [10] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of ICS*, June 2002.
- [11] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of MMCN*, January 2002.
- [12] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of SIGCOMM*, August 2001.
- [13] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of ICDE*, March 2003.