HRS: A Hybrid Replication Strategy for Exhaustive P2P Search

Hanhua Chen[1], Hai Jin[1], Xucheng Luo[2], and Zhiguang Qin[2]

[1] Huazhong University of Science and Technology, Wuhan, 430074, China
[2] University of Electronic Science and Technology of China, Chengdu, 6110054, China
hjin@hust.edu.cn

Abstract. Successful search and versatile query support are two important requirements for peer-to-peer (P2P) search applications. Replication strategy is an effective approach to improve the search performance of unstructured P2P systems. However, existing replication strategies either adapt only to popular queries or incur excessive replication cost for unpopular queries. In this work, we propose HRS, a hybrid replication strategy to improve the search performance of unstructured P2P networks. By combining a query popularity independent strategy with the square-root strategy, HRS can effectively and efficiently handle both kind of queries, popular or not. We evaluate this design through mathematical proof and comprehensive simulations. Results show that HRS outperforms existing replication-based search paradigms in terms of search performance and resource consummation.

1 Introduction

Since the emergence of peer-to-peer (P2P) [1,2,3] file sharing systems, such as Napster [4] and Gnutella [5], millions of users started to harness the desired data on the Internet with P2P tools. Recently, large scale P2P information sharing applications such as DistriWiki [6], Decentralized Wiki engine [7], and Peer-to-Peer Web [8,9] have attracted much attention. For this kind of application, both successful search and versatile query language support are important requirements. To guarantee successful search, unstructured P2P systems need exhaustive search techniques, where each $< item, query >$ (In this paper, we use "item", "data", "file", and "file metadata" interchangeably) pair can be evaluated with high probability and at low cost. Since network items can be in a heterogeneous format, such as html, XML, and other complex web objects, versatile query styles, such as keyword matching and XQuery [10], are preferable for a system design.

Current P2P systems mainly use three search schemes: flooding-based searching [11,1], Distributed Hash Table (DHT) looking up [12,13,14], and hybrid P2P searching [15,16,17]. In the first scheme, queries are flooded into an unstructured P2P network, suffering from excessive network traffic. DHT maintains a global index for item locating, guaranteeing a perfect successful rate while suffering from the problem of "exact match". Although some extensions based on

J. Cao et al. (Eds.): NPC 2008, LNCS 5245, pp. 173–184, 2008.

DHT are proposed to support complex queries, existing schemes incur unacceptable communication overheads [18]. Based on the observation that flooding is efficient for popular items while DHT is more suitable for rare items, Hybrid P2P [15,16,17] search schemes are proposed. A hybrid P2P network combines the unstructured protocol with the DHT global index, and performs a query by either flooding or DHT looking up according to the item's popularity.

Unstructured P2Ps are naturally the best candidate for supporting versatile queries because the matching operations can be evaluated at the nodes that store the relevant items. The first unstructured P2P protocol, Gnutella, is not scalable due to the adoption of flooding query scheme. An efficient approach to improve the search performance of unstructured P2Ps is to utilize replication strategies. The existing replication strategies can be divided into two categories. The first type is the query popularity aware strategies. The number of replicas is determined by the query's popularity. Existing research [11] claimed that the square-root replication strategy has the optimal expected search size (ESS), which is the average number of random probes required to solve a query. However, this strategy is inefficient for solving "insoluble queries", the queries for rare and non-existent items. For non-existent items, the query stop rule is crucial for reducing the search cost. Obviously, it can not guarantee the query to be searched exhaustively. The second type is independent of the popularity of a query, such as RWPS [19], Bubblestorm [20], and RandRep [21]. This kind of strategy deploys an optimal number of item replicas randomly in a P2P network to achieve probabilistically exhaustive search, without exploiting the query's popularity to reduce the query overhead. For example, in Bubblestorm, each item, popular or not, has the same number of replicas, which is determined by the network size.

The key issues for replication-based probabilistically exhaustive search in unstructured P2P networks is how to estimate the optimal number of replicas and disseminating the replicas optimally throughout the network. In this paper, we propose HRS, a hybrid replication strategy to improve the search performance of unstructured P2P networks. By combining a query popularity independent strategy with the square-root strategy, HRS can effectively handle queries for both popular and rare items. We conduct comprehensive simulations to evaluate this design. Results show that HRS outperforms existing techniques in terms of search performance and search cost.

The remainder of the paper is organized as follows. In Section 2, we review related work. The model and the problem statement are given in Section 3. Section 4 presents the design of HRS. We evaluate the performance of HRS in Section 5. We conclude in Section 6.

2 Related Work

Without centralized index servers, nodes in a decentralized P2P system have to cooperate with each other to perform a search for desired data items. Existing systems utilize replication strategies to improve the search performance. Existing replication strategies in unstructured P2P networks can be divided into

two categories, the query popularity aware replication approach and the query popularity independent replication strategy.

2.1 Query Popularity Aware Replication

In this kind of strategy, the number of replicas is related to the query rates. Let r_i denote the number of replicas of item i.The sum of the replica amounts is $R = \sum_{i=1}^{m} r_i$. Let q_i denote the query rate of item i, which is the fraction of all queries that are issued for item i. The number of replicas in this strategy is $r_i = f(q_i)$. Two natural strategies among existing schemes are uniform and proportional strategies, while in the uniform replication strategy, all items are equally replicated, that is $r_i = R/m$. In the proportional replication strategy, the number of replicas is proportional to the query rates, that is $r_i = R \times q_i$. Cohen et al. [11] have studied the two query-rate based strategies. Their analysis results show that the above two strategies are not optimal as to the expected number of random probes (ESS). Another result is that the above two strategies have the same ESS and any strategies between them are better than them as to the ESS. Cohen et al. then propose the square-root replication (SRR), where the number of replicas is proportional to the square-root of the query rates. In SRR, the number of replicas is $r_i = \lambda\sqrt{q_i}$, where $\lambda = R/\sum_{i=1}^{m}\sqrt{q_i}$. They also prove that SRR is optimal as to ESS. In short, as to the expected search size, uniform strategy is the same as proportional strategy. The square-root replication strategy achieves optimal expected search size. The average search size of uniform replication strategy and proportional replication strategy is given by

$$E\left[T_{uniform}\right] = E\left[T_{proportional}\right] = \frac{Nm}{R} \tag{1}$$

The ESS of SRR is given by

$$E\left[T_{optimal}\right] = \frac{R}{N}(\sum_{i=1}^{m}\sqrt{q_i})^2 \tag{2}$$

Although square-root replication can achieve optimal expected search size, it is only practical for "soluble queries". In SRR, the "soluble queries" are queries which can be solved within the given maximum search size. However, defining the maximum search size is not easy. For items with a small number of replicas, to guarantee exhaustive search, the number of random probes is very big. SRR refers to queries for these kinds of items as "insoluble queries", which can not be solved efficiently by SRR. Furthermore, how to divide popular and unpopular queries is also not clear. Thus, exhaustive search can not be guaranteed with high probability at low cost in this scheme.

2.2 Query Popularity Independent Replication

Recently, the query popularity independent replication strategy has attracted much attention. All items are equally replicated regardless of the popularity

of the related queries. For file search, queries are replicated to some random nodes. The well-chosen parameters guarantee the collision of item replica and query with high probability. This idea is inspired by the birthday paradox [22]. However, since item and query replications are two independent processes, the birthday paradox can not be directly used to design the related parameters.

RWPS [19] firstly propose this kind of replication in a Gnutella network. To implement the installation of replicas, RWPS employs random walk to sample some random nodes. RWPS can guarantee exhaustive search with high probability. However, random walk is not fault-tolerant, a failed node in the path could reduce replica amount. Thus, the search success probability can not be guaranteed. On the other hand, random walk has long latency.

To overcome the shortcoming of RWPS, Terpstra et al. propose Bubblestorm [20] to achieve probabilistic and exhaustive search. Bubblestorm employs random multi-graph to connect peers. The related joining and leaving protocols are designed to keep the attributes of random multi-graph. To improve the performance of message propagating, they also design a new algorithm-bubblecast to distribute replicas and queries, which is the combination of both random walk and flooding.

RandRep [21] is another implementation of query rate independent replication. In this scheme, a lightweight DHT is employed to support network size estimation and random node selection. All items have equal numbers of replicas. To guarantee the search success with high probability, the number of replicas is carefully determined. Let r and q denote the numbers of item replicas and query replicas, respectively. If $rq \geq N\left((1+\varepsilon\ln N)+\sqrt{(1+\varepsilon\ln N)^2-1}\right)$, the probability of two kinds of replicas encountering in at least one node is $P \geq 1-N^{-\varepsilon}$, where $\varepsilon > 0$ is a constant. For current P2P network size, it is suitable that the value of rq is $N(2+\ln N)$. If the traffic of each query and item replica is the same, the value of r and q is $\sqrt{N(2+\varepsilon\ln N)}$, that is to say $r = q = O(\sqrt{N\ln N})$.

Query popularity independent strategy can achieve exhaustive search with high probability. Since the query popularity is not considered in this strategy, it is not optimal for popular queries in terms of search cost. The drawback of such strategy is very clear. Uniformly replicating all items including the infrequently queried ones, is inefficient. On the contrary, for the popular queried items, a small number of query replicas can efficiently reduce the overall cost of the search system.

3 System Model and Problem Statement

The model is related to the overlay network, item replicas, and queries. The network is composed of N nodes. Several approaches for obtaining the network size N have been proposed, for example [23,24]. There are m items shared in the network. Each item, i, is replicated at r_i sites. The vector of replica amounts is $(r_1, r_2, \ldots, r_m)$. The sum of the replica amounts is $R = \sum_{i=1}^{m} r_i$. The query rate vector for items is $(q_1, q_2, \ldots, q_m)$, where q_i is the query-rate for item i, which

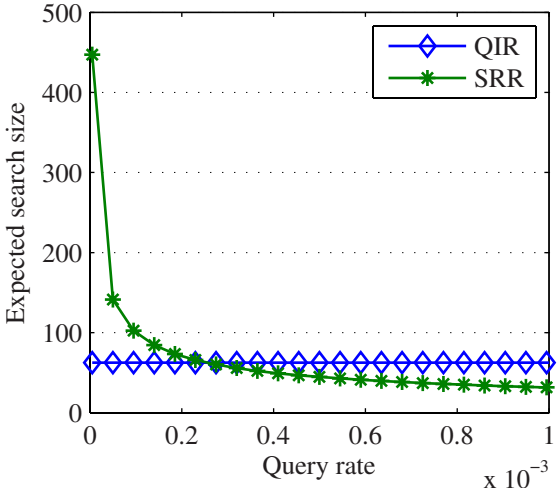

Fig. 1. Search size comparison for QIR and SRR

is the fraction of all queries issued for item i. The rate q_i is normalized, that is $\sum_{i=1}^{m} q_i = 1$. If all items are equally replicated, the number of replicas is denoted as r. Correspondingly the query amounts for specific items are equal, which is denoted as q. In this kind of P2P sharing network, we consider the random search method adopted in [11]. This random search method sends query messages to random nodes in the network until the query is answered or the search reaches the maximum size. There are several approaches to achieve the random nodes, for example, the combination of random multi-graph with bubblecast in Bubblestorm [20] and the RandRep scheme which combines a lightweight DHT with an unstructured P2P overlay to address random peer sampling [21].

In this paper, we propose HRS, a hybrid replication strategy to improve the search performance of unstructured P2P networks. HRS performs an exhaustive search, which probabilistically guarantees that the application's query evaluator runs on a peer containing a replica of the related item. By combining a query popularity independent strategy with the square-root strategy, HRS can effectively handle queries for items, popular or not. We address the following problem. What is the efficient and practical replication strategy for a query issued in unstructured P2P networks regardless of how popular it is?

4 HRS Replication

For real systems, popular queries, unpopular queries, and even queries for non existent items should all be considered. Square-root replication strategy is just practical for popular queries. As for unpopular queries and queries for non existent items, query rate independent replication (QIR) strategy is better than square-root replication strategy. Figure 1 illustrates the search size comparison

of QIR and SRR in a 50,000 node network. At the query rate about 0.27, QIR and SRR have the same search size. Prior to that, QIR has a smaller search size. After that, SRR performs better. Therefore, it would be reasonable to combine SRR and QIR to meet the requirements of real systems. Based on the observation, we propose HRS, which is a combination of SRR and QIR. The challenge here is how to implement the effects of both SRR and QIR.

4.1 HRS Strategy

Replication strategy is an allocation of replicas for each item. The objective is to improve the related performance metrics. In order to consider both query and item popularity, the following allocation function is given.

$$r_i = f(N, q_i) \tag{3}$$

For an item with very small q_i, to guarantee exhaustive search, the fraction of all nodes which have a replica of item i should be bigger than a given threshold. Under this condition, this item can be searched within a given maximum search size with high probability. If the maximum search size is reached while the item is not found, it is the case that there are no such item in the network with high probability. In the QIR strategies, each item has an equal number of replicas to guarantee the exhaustive search. Thus, this value can be treated as the minimization of the number of replicas in HRS. That is given by

$$r_{min} = \sqrt{N\,(2 + \ln N)} \tag{4}$$

On the other hand, the query rate should be considered. Obviously, we should incorporate the square-root strategy in HRS. There is a critical point of the query rate. At this point, we have the below equation.

$$\sqrt{N\,(2 + \ln N)} = \frac{R \times \sqrt{q_i}}{\sum_{j=1}^{m} \sqrt{q_j}} \tag{5}$$

Then, the critical point of the query rate is

$$q_c = \frac{N(2 + \ln N)(\sum_{j=1}^{m} \sqrt{q_j})^2}{R^2} \tag{6}$$

Therefore, for items with query rate bigger than q_c, the square-root strategy is employed. Other items should have r_{min} replicas.

According to above analysis, the HRS replica allocation function is given by

$$r_i = \begin{cases} \sqrt{N(2 + \ln N)} & \text{where } q_i < q_c \\ \frac{R \times \sqrt{q_i}}{\sum_{j=1}^{m} \sqrt{q_j}} & \text{where } q_i \geq q_c \end{cases} \tag{7}$$

4.2 Implementation of HRS

HRS employs reactive operations to replicate items. Two actions trigger the replication operation. The first action is node joining. At this point, there are no replicas of items shared by this node. The corresponding query rates are zero, which is obviously smaller than the critical point p_c. Therefore, a minimum number of replicas should be distributed in the network. The new node replicate their items to r_{min} random nodes. This operation guarantees an exhaustive search with high probability. The second action is item querying. A query to item F may succeed or fail. If $s \leq \sqrt{N(2+\ln N)}$ probes are required to get the answer, s replicas of item F are added into the network after the search. Otherwise, the query fails, which implies that the queried item is non existent. This operation is used to maintain the square-root replication strategy.

We adopt random walk to implement HRS. Random walk is an efficient approach to solve problems in random networks, for example, resource search [25], overlay construction [26]. Another important functionality of random walk is random node sampling which is the base for random replication. As to the sampling efficiency, it is optimal for sampling r random nodes at the cost of r messages. Unstructured P2P networks have two important attributes related to node sampling. The first is that the topology is modeled as a random graph. Usually, the Poisson random graph and power-law random graph [27] are used to model unstructured P2P networks. The second attribute is that P2P systems are dynamic system. Nodes can only maintain their neighborhood information efficiently. Due to the two attributes, random walk is a reasonable approach for sampling random node subset. HRS employs random walk to implement optimal random node subset sampling.

For arbitrary node $v_i \in V$, $\Gamma(v_i)$ is the set of nodes which connect to node v_i and $d_i = |\Gamma(v_i)|$ is the degree of node v_i. Random walk on a graph is a sequence of nodes, for example $v_0, v_1, \cdots, v_i, v_j, \cdots$. If the position is v_i at time t, the probability of reach vertex v_j at time $t+1$ is:

$$p_{ij} = \begin{cases} 1/d_i \text{ if } i \neq j \text{ and } j \in \Gamma(v_i) \\ 0 \quad \text{otherwise} \end{cases} \tag{8}$$

Random walk on a graph is a Markov chain. The corresponding probability transition matrix is $P = \{p_{ij}\}$. The initial distribution is denoted as $\pi(0)$. The distribution at time t is $\pi(t)$. If the Markov chain is irreducible, finite, and aperiodic, this Markov chain has unique stationary distribution [28]. Random walk on P2P networks is corresponding to this kind of Markov chain. If the graph is regular, random walk reaches each node with equal probability after the Markov chain converging. However, P2P networks are not regular graph, simple random walk can not obtain uniform sampling. Two alternatives are proposed to address uniform sampling in non-regular graph. The first is the Maximum Degree random walk (MD), in which the graph is converted to regular graph by adding self-loop to low degree nodes. The second is the Metropolis-Hasting random walk (MH). The transition probability of MH is given by

$$p_{ij} = \begin{cases} 1/\max(d_i, d_j) & \text{if } i \neq j \text{ and } j \in \Gamma(i) \\ 1 - \sum\limits_{j \in \Gamma(i)} p_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

In HRS, MH is used to sample nodes.The time from the initial distribution to the stationary distribution is called mixing time. Although MD is very simple, its mixing time is bigger than that of MH [29]. The existing random walk based node sampling algorithms are one random node each time. For sampling a random subset, it is straightforward to sample multiple times. Due to excessive message overhead and big latency, however, this strategy is not practical. According to recently research results, we find that the efficiency of random walk-based sampling can be improved. For MH random walk on a random graph, continuous s steps obtain s random nodes. After random walk on a random graph converges, the successive s steps obtain s random nodes [26]. For random walk start from node A, it reaches node B after convergence. As to A, B is a random node. According to the reverse random walk path principle [30], A is also a random node for B. Since s step random walk from B is a sampling of s random nodes, s step random walk from A is also a sampling of s random nodes. That is to say, s step random walk from arbitrary node obtains s random nodes.

HRS can efficiently solve queries for any items. In HRS, the baseline of the replica amount for any item is r. This setting guarantees all queries can be solved or ended with at most $\left\lceil \sqrt{N(2 + \ln N)} \right\rceil$ probes. For popular queries, the reactive replication operation inserts additional replicas into the network. As the replica amount increases, the search size decreases. However, the search cost for each query does not increase.

5 Performance Evaluation

In this section, we present the performance metrics, experimental setup, and performance evaluation in our simulations.

5.1 Simulation Methodology

Many metrics are related to the search performance, for example, search success probability, search size, search traffic, search cost, and query delay. In HRS, the replica amount guarantees the search success with high probability. As for queries for non existent items, the search stop rule is the maximum search size. Therefore, it is not necessary to measure the search success probability. In the design, the random probe is used to search items, which is also adopted in the square-root replication strategy [11]. In the evaluation of HRS, we are mainly concerned with search size and search cost. The search size is the number of random probes until the termination of the search. The search cost is averaged. For some item, the number of replicas is x. The number of queries for this item is y. If the sum of probes is p, the search cost is $(x + p)/y$.

In the simulation, networks with 10,000, 300,000, 500,000, 700,000, 900,000, and 110,000 nodes are constructed. Initially, each item is treated as a new item and uniformly replicated at $\left\lceil \sqrt{N(2+\ln N)} \right\rceil$ random nodes. Queries for these items are put into the network to trigger the replication. The measurements of real P2P systems show that the popularity of queries follow a Zipf-like distribution [31]. For a query with popularity rank i, the number of this query is proportional to $i^{-\alpha}$. In the simulation, the parameter α is 0.6 according to the measurements. We generate the queries according to the Zipf distribution and insert these queries into the networks.

5.2 Results

Firstly, we implement the QIR strategy. Each item is equally replicated, then, different queries are issued. Each query is repeated 10,000 times and the averages are calculated. The result is illustrated in Fig. 2. Compared with the network size, the number of replicated items is small. The average number of probes for search success is also very small. For a network with 110,000 nodes, this value is about 100.

The search performance of HRS is shown in Fig. 3. The results from 50,000, 70,000, and 90,000 node networks are illustrated. In the simulation, each query is repeated 10,000 times and the average is calculated. As the number of queries increases, the number of probes decreases rapidly. This distribution shows that the query popularity can be exploited to reduce the search size.

Figure 4 is the search cost comparison of HRS and QIR. The network size is 100,000. When the query is not popular, HRS and QIR have almost the same search cost. As the query popularity increases, the search cost gap of HRS and QIR becomes larger. Furthermore, we study the sum of search cost under Zipf query popularity distribution. The results are illustrated in Fig. 5 in which the parameter α is 0.6. The search cost of HRS is less than that of QIR.

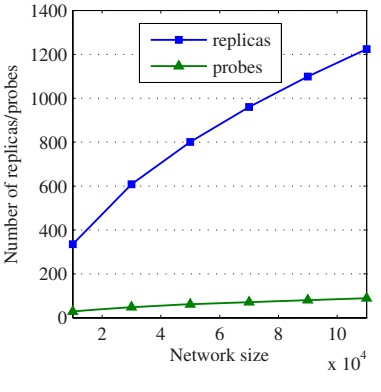

Fig. 2. The number of probes/replicas for different network sizes

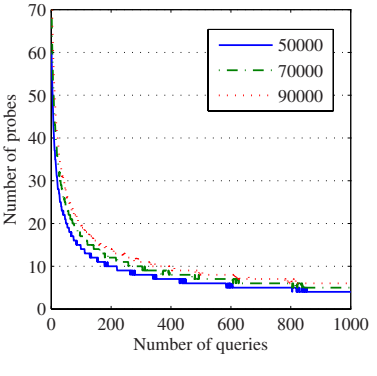

Fig. 3. The search size of HRS in networks with different sizes

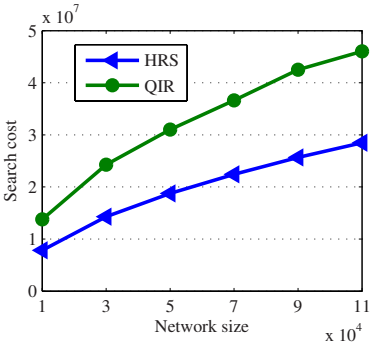

Fig. 4. The search cost comparison of HRS and QIR

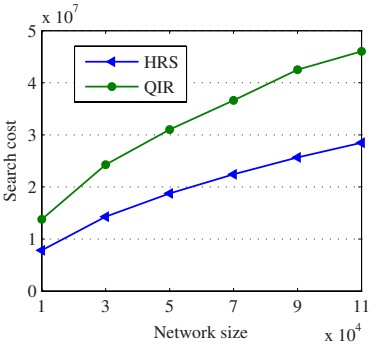

Fig. 5. The search cost comparison with Zipf query popularity distribution ($\alpha = 0.6$)

6 Conclusions

To address the issue of exhaustive search in unstructured P2P networks, we propose HRS, a hybrid replication strategy to improve the search performance of unstructured P2P networks. By combining a query popularity independent strategy with the square-root strategy, HRS can effectively handle queries, popular or not. We evaluate this design through mathematical proof and comprehensive simulations. Results show that HRS can achieve a small search size at a reasonable search cost. Therefore, HRS is a practical replication strategy for real P2P systems.

Acknowledgements

This work was supported by National Science Foundation of China (NSFC) under grants No.60433040 and No. 60473090, NSFC/RGC Joint Research Foundation under grant No.60731160630, and National 973 Key Basic Research Program under grant No.2003CB317003.

References

1. Liu, Y., Liu, X., Xiao, L., Ni, L.M., Zhang, X.: Location-aware topology matching in p2p systems. In: Proceedings of IEEE INFOCOM 2004, Hong Kong, China. IEEE, Los Alamitos (2004)
2. Cao, J., Liu, F.B., Xu, C.Z.: P2pgrid: integrating p2p networks into the grid environment. Concurr. Comput. Pract. Exper. 19(7), 1023–1046 (2007)
3. Li, M., Lee, W.C., Sivasubramaniam, A.: Semantic small world: An overlay network for peer-to-peer search. In: Proceedings of IEEE ICNP 2004, Washington, DC, USA, pp. 228–238. IEEE Computer Society Press, Los Alamitos (2004)
4. Saroiu, S., Gummadi, K., Gribble, S.: Measuring and analyzing the characteristics of napster and gnutella hosts. Multimedia Systems Journal 9(2), 170–184 (2003)

5. Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N., Shenker, S.: Making gnutella-like p2p systems scalable. In: Proceedings of ACM SIGCOMM 2003, Karlsruhe, Germany, pp. 407–418. ACM, New York (2003)
6. Morris, J.C.: Distriwiki: a distributed peer-to-peer wiki network. In: Proceedings of ACM ISW 2007, Quebec, Canada, pp. 69–74. ACM, New York (2007)
7. Urdaneta, G., Pierre, G., Steen, M.v.: A decentralized wiki engine for collaborative wikipedia hosting. In: Proceedings of WEBIST 2007, Barcelona, Spain, pp. 855–858. Springer, Heidelberg (2007)
8. Chen, H., Jin, H., Wang, J., Liu, Y., Ni, L.M.: Efficient multi-keyword search over p2p web. In: Proceedings of ACM WWW 2008, Beijing, China, pp. 989–997. ACM, New York (2008)
9. Deshpande, M., Amit, A., Chang, M., Venkatasubramanian, N., Mehrotra, S.: Flashback: A peer-to-peer web server for flash crowds. In: Proceedings of IEEE ICDCS 2007, Toronto, Ontario, Canada, p. 15. IEEE, Los Alamitos (2007)
10. Chamberlin, D.: Xquery: a query language for XML. In: Proceedings of ACM SIGMOD 2003, San Diego, California, USA, p. 682. ACM, New York (2003)
11. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. In: Proceedings of ACM SIGCOMM 2002, Pittsburgh, PA, USA, pp. 177–190. ACM, New York (2002)
12. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications. In: Proceedings of ACM SIGCOMM 2001, San Diego, California, USA, pp. 149–160. ACM, New York (2001)
13. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proceedings of ACM SIGCOMM 2001, San Francisco, California, USA, pp. 161–172. ACM, New York (2001)
14. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: a resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications 22(1), 41–53 (2004)
15. Loo, B.T., Huebsch, R., Stoica, I., Hellerstein, J.M.: The case for a hybrid p2p search infrastructure. In: Voelker, G.M., Shenker, S. (eds.) IPTPS 2004. LNCS, vol. 3279, pp. 141–150. Springer, Heidelberg (2005)
16. Chen, H., Jin, H., Liu, Y., Ni, L.M.: Difficulty-aware hybrid search in peer-to-peer networks. IEEE Transactions on Parallel and Distributed Systems (TPDS) (2008)
17. Zaharia, M., Keshav, S.: Gossip-based search selection in hybrid peer-to-peer networks. In: Proceedings of IPTPS 2006, Santa Barbara, CA, USA (2006)
18. Li, J., Loo, B.T., Hellerstein, J., Kaashoek, F., Karger, D.R., Morris, R.: On the feasibility of peer-to-peer web indexing and search. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 207–215. Springer, Heidelberg (2003)
19. Ferreira, R.A., Ramanathan, M.K., Awan, A., Grama, A., Jagannathan, S.: Search with probabilistic guarantees in unstructured peer-to-peer networks. In: Proceedings of P2P 2005, Konstanz, Germany, pp. 165–172. IEEE, Los Alamitos (2005)
20. Terpstra, W.W., Kangasharju, J., Leng, C., Buchmann, A.P.: Bubblestorm: Resilient, probabilistic, and exhaustive peer-to-peer search. In: Proceedings of ACM SIGCOMM 2007, Kyoto, Japan, pp. 49–60. ACM, New York (2007)
21. Luo, X., Qin, Z., Han, J., Chen, H.: Dht-assisted probabilistic exhaustive search in unstructured p2p networks. In: Proceedings of IEEE IPDPS 2008, Miami, Florida, USA. IEEE, Los Alamitos (2008)
22. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)

23. Massouli, L., Le Merrer, E., Kermarrec, A.M., Ganesh, A.: Peer counting and sampling in overlay networks: Random walk methods. In: Proceedings of ACM PODC 2006, Denver, CO, USA, pp. 123–132. ACM, New York (2006)
24. Kostoulas, D., Psaltoulis, D., Gupta, I., Birman, K., Demers, A.: Decentralized schemes for size estimation in large and dynamic groups. In: Proceedings of IEEE NCA 2005, Cambridge, MA, USA, pp. 41–48. IEEE, Los Alamitos (2005)
25. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceedings of ACM ICS 2002, pp. 84–95. ACM, New York (2002)
26. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks. In: Proceedings of IEEE INFOCOM 2004, Hong Kong, China. IEEE, Los Alamitos (2004)
27. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509 (1999)
28. Lovasz, L.: Combinatorics. Paul Erdos is Eighty 2(2), 353–398 (1996)
29. Awan, A., Ferreira, R.A., Jagannathan, S., Grama, A.: Distributed uniform sampling in unstructured peer-to-peer networks. In: Proceedings of HICSS 2006, Washington, DC, USA, p. 223.3. IEEE, Los Alamitos (2006)
30. Bar-Yossef, Z., Friedman, R., Kliot, G.: Rawms - random walk based lightweight membership service for wireless ad hoc networks. ACM Trans. Comput. Syst. 26(2), 1–66 (2008)
31. Sripanidkulchai, K.: The popularity of gnutella queries and its implications on scalability. In: Oram, A. (ed.) The O'Reilly Peer-to-Peer and Web Services Conference. O'Reilly, Sebastopol (2001)