On the Feasibility of Unstructured Peer-to-Peer Information Retrieval

H. Asthana, Ruoxun Fu and Ingemar J. Cox
h.asthana,r.fu,ingemar@cs.ucl.ac.uk

Department of Computer Science, University College London, Gower St., London
WC1E 6BT, UK

Abstract. We consider the feasibility of web-scale search in an unstructured peer-to-peer network. Since the network is unstructured, any such search is probabilistic in nature. We therefore adopt a probably approximately correct (PAC) search framework. The accuracy of such a search is defined by the overlap between the set of documents retrieved by a PAC search and the set of documents retrieved by an exhaustive (deterministic) search of the network. For an accuracy of 90%, we theoretically determine the number of nodes each query must be sent to for three distributions of documents in the network, namely uniform, proportional and square root. We assume that the query distribution follows a power law and investigate how performance is affected by the scale factor. For various configurations, we estimate the global and local network traffic induced by the search. For a network of 1 million nodes, a query rate of 1000 queries per second, and assuming each node is capable of indexing 0.1% of the collection, our analysis indicates that the network traffic is less that 0.07% of global internet traffic.

1 Introduction

P2P networks can be generally categorized into two classes, namely structured and unstructured networks. Structured networks, typically based on distributed hash tables (DHTs), bind data to designated locations within the network. The advantage of a structured architecture is that the query latency, proportional to the number of nodes a query must visit, is $O(\log n)$ where n is the number of nodes in the network. However, multi-term queries can consume considerable bandwidth as nodes need to exchange information regarding the sets of documents containing each term [1]. Additional bandwidth is needed to maintain the binding, and can grow very quickly in the face of dynamic membership (churn), which can in turn saturate the network. Further concerns have been raised in [2]. In particular, distributed hash tables are particularly susceptible to adversarial attack [3].

Unstructured networks exhibit no such binding between data and nodes. As such, they are much less affected by churn, and are generally more resistant to adversarial attack. However, since a particular document being sought by a user can be anywhere in the network, the only way to guarantee searching the entire collection is to exhaustively query all nodes in the network. This is, of course, impractical. To be practical, any search must only query a relatively small subset of nodes in the network. Thus, search in an unstructured P2P network is necessarily probabilistic.

One of the earliest attempts to estimate the feasibility of web search in peer-to-peer networks was undertaken by Li *et al.* [1] which examined the practicality of web search based on a structured P2P network indexing 3 billion documents

Published in the proceedings of the 3rd International Conference on the Theory of Information Retrieval (ICTIR), 2011

and concluded that the bandwidth required was still an order of magnitude greater than was practical at the time. Even after applying various optimization techniques, the estimated query size (i.e. the total traffic generated in issuing and answering a query) was still found to be 6 MB per query. Whilst this (relatively high) communication cost would now be feasible, as we discuss later, the analysis in [1] does not account for expected churn rates in observed peer-to-peer systems which can significantly increase the required bandwidth in structured networks for keeping the DHT up-to-date. Zhong *et al.* [4] investigated the effectiveness of various indexing strategies in structured peer-to-peer networks using 3.7 million queries. However, the authors did not investigate unstructured peer-to-peer networks and did not address the concerns, previously mentioned, regarding structured peer-to-peer networks. Yang *et al.* [5] compared the performance of keyword search in structured, super-peer, and unstructured peer-to-peer network by downloading the documents from 1,000 web sites and allocating each peer to "host" one web site. The conclusion of the study is that the performance of the three network types is similar. However, since the documents are only present at one node, there is no *replication of documents* - a key concept which we explore in the next section.

To our knowledge, there have been no studies into whether it is possible to perform web search in unstructured peer-to-peer networks which takes into account document replication with high probabilities of finding the relevant document(s).

The main contributions of this paper are

- a theoretical analysis that predicts the number of nodes that must be queried in order to guarantee an expected accuracy of 90% for three different document replication policies
- estimation of the corresponding communication bandwidth required, concluding that probabilistic search in an unstructured peer-to-peer network where nodes issue queries in volume comparable to commercial search engines will consume no more than 0.07% of the global internet traffic.

In this paper, the feasibility of web scale search in unstructured peer-to-peer networks is based on communication cost. However, we acknowledge that other factors are also of concern, e.g. latency and security. Latency is directly proportional to the number of nodes queried. Although it is part of our future research, we do not address latency in this paper, and refer the reader to [6, 7], which discusses the optimal network topology to reduce latency. The security of an unstructured peer-to-peer network is outside the scope of the paper. However, initial investigations, not reported here, suggest that security is better than that for structured architectures based on distributed hash tables.

In Section 2 we review prior work on probabilistic search in an unstructured P2P network. In Section 3 we extend the probably approximately correct search architecture, which is a recently proposed unstructured P2P search framework, to incorporate non-uniform replication strategies. In Section 4 we calculate the communication cost of web search in unstructured P2P networks based on the theoretical results of Sections 2 and 3. We conclude in Section 5.

2 Probabilistic Search

Probabilistic storage and search in an unstructured P2P network can be modeled as follows. Given a set of n nodes in the network, we assume that the object of interest is stored on a random subset of r nodes. A query is issued to a random subset of z nodes. We are interested in the probability that the two subsets have a non-empty intersection, as this implies a successful search for that object. This theoretical foundation is directly adopted in prior work which focuses on retrieval of files stored in the network based on queries that contain terms that only appear in the file names. Information retrieval is broader than this, as the index, and associated queries, contain terms present not just in the file name, but also terms present within the file (document) itself. As such, matching of queries to documents is more ambiguous, and it is therefore necessary to provide a set of documents, usually ranked by relevance. Nevertheless, this model is appropriate for the class of information retrieval problems referred to as *known-item* search. And, the probabilistic model can be extended to encompass other information retrieval requirements, as discussed in Section 2.3.

2.1 Ferreira et al's Model

Early work on probabilistic search in unstructured P2P networks has its origins in the study of probabilistic quorum systems [8] to improve the availability and efficiency of replicated systems. Ferreira *et al.* [9] proposed the use the probabilistic quorum model to describe search in an unstructured P2P network. Given n nodes in the network, an object is replicated $\gamma\sqrt{n}$ times onto a random subset of nodes. A query is also sent to a random subset of $\gamma\sqrt{n}$ nodes. It can then be shown that the probability of finding the desired object of the query is at least $1 - e^{-\gamma^2}$. Clearly as γ increases, the object is replicated over more nodes and the probability of finding the object therefore increases.

2.2 Cohen and Shenkers' Model

The previous analysis assumed that an object/document is uniformly randomly replicated across nodes in the network. Other replication strategies are also possible. Cohen and Shenker [10] provided both a theoretical and empirical analysis of such. Here it is assumed that the n nodes in the P2P network each have capacity ρ, i.e. ρ is the number of files each node can store. Let $R = n\rho$ denote the total capacity of the system. It is further assumed that there are m unique files stored in the P2P system and that each file i is replicated on r_i random nodes. Obviously, $\sum_i r_i = R$, and $mr = R$ if $r_i \equiv r$ is a constant. Let $p_i = \frac{r_i}{R}$ be the fraction of the total system capacity allocated to file i. Finally, let q_i be the normalized query popularity for the ith file. Thus

$$\sum_{i=1}^{m} q_i = 1$$

The search size, z_i, of file i, is defined as the number of nodes searched in response to a query q_i to find file i. Of course, the search size will depend very much on the search strategy used. In [10] a random probing model is assumed, i.e. each of many probes randomly selects a node in the network. Thus, each probe has a probability $\frac{r_i}{n}$ of finding the requested file i, and a probability $1 - \frac{r_i}{n}$ of not finding the file. The search size z_i is simply a random variable drawn from a geometric distribution

$$P(z_i) = (1 - \frac{r_i}{n})^{z_i - 1}\frac{r_i}{n}$$

The average search size for file i is

$$\mu_z(i) = \frac{n}{r_i}$$

and the expected search size, μ_z, of all m files is

$$\mu_z = \sum_i q_i \times \mu_z(i) = n\sum_i \frac{q_i}{r_i} = n\sum_i \frac{q_i}{Rp_i} = \frac{n}{R}\sum_i \frac{q_i}{p_i} \tag{1}$$

For a uniform replication strategy, $r_i \equiv r$ is a constant and $mr = R$. Thus, the expected search size with uniform replication, μ_z^u, is

$$\mu_z^u = n \sum_i \frac{q_i}{r} = \frac{n}{r} \sum_i q_i = \frac{m}{\rho} \tag{2}$$

Two alternatives to a uniform replication strategy are also considered. A proportional replication strategy replicates content based on its popularity, i.e. proportional to the number of queries requesting it. Perhaps surprisingly, such a replication strategy results in the same expected search size as the uniform replication. For proportional replication strategy, $r_i = Rq_i$ and the expected search size μ_z^p is

$$\mu_z^p = n \sum_i \frac{q_i}{Rq_i} = \frac{nm}{R} = \frac{m}{\rho}$$

In effect, while popular documents will be found by querying fewer nodes than for a uniform replication strategy, this is balanced by the need to visit far more nodes in order to find unpopular documents.

To minimize the expected search size, we would like to minimize $\sum_i \frac{q_i}{p_i}$ in Equation (1). Solving this optimization problem [10], we have $\frac{p_i}{p_m} = \frac{r_i}{r_m} = \frac{\sqrt{q_i}}{\sqrt{q_m}}$. Thus, $r_i = \lambda\sqrt{q_i} = \frac{R}{\sum_i \sqrt{q_i}}\sqrt{q_i}$. This is the square root replication strategy which produces the optimal expected search size given by

$$\mu_z^s = n \sum_i \frac{q_i}{\lambda\sqrt{q_i}} = \frac{1}{\rho}(\sum_i \sqrt{q_i})^2 \tag{3}$$

Typically, the query distribution follows a power law [11], i.e. $q_i = \frac{1}{c} i^{-\alpha}$ where c is the normalization constant. In this case, Equation (3) becomes

$$\mu_z^s = \frac{1}{\rho c}(\sum_i \frac{1}{i^{\alpha/2}})^2 \tag{4}$$

Analysis of publicly available logs from AOL, as well as logs of a commercial search engine made available to us, indicate that the value of α ranges from 0.8 to 1.0.

2.3 Probably Approximately Correct Search

The previous work on randomized search looked at the expected search length necessary to find a specific document. Assuming a query is sent to a constant number of nodes, z, we can also ask what the probability of finding a document is. This, and related questions, are addressed in recent papers on probably approximately correct (PAC) search [12, 13]. The PAC architecture considers both an acquisition and a search stage. In this paper we only consider the search stage.

During the search stage, a query is sent to z machines, and the results returned by the different machines are consolidated and then displayed to the user. If we are searching for a single, specific document d_i, then the probability of retrieving this document is given by

$$P(d_i) = 1 - (1 - \frac{\rho}{m})^z \tag{5}$$

In information retrieval, it is more common to be interested in the top-k retrieved documents. In this case, the correctness of a PAC search is measured by *retrieval* *accuracy*. If $\mathcal{D}$ denotes the set of top-k documents retrieved when searching the full index, i.e. an exhaustive search, and $\mathcal{D}'$ the set of top-k documents retrieved when querying z nodes, then the retrieval accuracy, a, is defined as

$$a = \frac{|\mathcal{D} \cap \mathcal{D}'|}{|\mathcal{D}|} = \frac{k'}{k}$$

where k' denotes the size of the overlap of the two sets, i.e. $|\mathcal{D} \cap \mathcal{D}'|$.

The size of the overlap in the result sets, k' is a random variable drawn from a binomial distribution, and is given by

$$P(k') = (\genfrac{}{}{0pt}{}{k}{k'})P(d_i)^{k'}(1 - P(d_i))^{k-k'} \tag{6}$$

Since Equation (6) is a binomial distribution, the expected value of k' is $E(k') = kP(d_i)$ and the expected retrieval accuracy μ_e is

$$\mu_e = \frac{\mu_{k'}}{k} = \frac{k \times P(d_i)}{k} = 1 - \left(1 - \frac{\rho}{m}\right)^z, \tag{7}$$

If we assume that a document is, on average, replicated r times onto different nodes in the network, then the total storage of the network, R, satisfies $R = m \times r$. And the Equation (7) can be transformed into

$$\mu_e = 1 - \left(1 - \frac{\rho}{m}\right)^z = 1 - \left(1 - \frac{r}{n}\right)^z \tag{8}$$

3 Non-uniform replication in PAC

The original work on PAC, described above, assumed a uniform replication strategy for documents. Here, we extend the probabilistic analysis to the cases where the documents are replicated (i) in proportion to their popularity, and (ii) in proportion to the square root of their popularity, as discussed in [10].

In general, given a query distribution, we are interested in what replication strategy can yield the highest expected retrieval accuracy for all queries. Let $\mathcal{Q}$ denote the set of all queries, and let q_j denote the query rate for query j, such that $\sum q_j = 1$. The replication rate for document i is r_i. From Equations (5) and (8), we can get the probability of retrieving document i as

$$P(d_i) = 1 - \left(1 - \frac{r_i}{n}\right)^z \tag{9}$$

Let $\mathcal{D}_k(j)$ denote the set of top-k documents retrieved for a query q_j. Thus, the expected retrieval accuracy for the top-k documents, A_k, averaged over all queries is given by

$$A_k = \sum q_j \frac{\sum_{d_i \in \mathcal{D}_k(j)}(1 - (1 - \frac{r_i}{n})^z)}{k} \tag{10}$$

Now consider the case where we are only interested in the top document, i.e. top-1 ($k = 1$), as in [10]. Thus, from Equation (10) we have

$$A_1 = \sum q_j(1 - (1 - \frac{r_j}{n})^z) \tag{11}$$

For a proportional replication strategy, where $r_j = Rq_j$, the expected accuracy A_1^p is then given by

$$A_1^p = \sum q_j(1 - (1 - \rho q_j)^z) \tag{12}$$

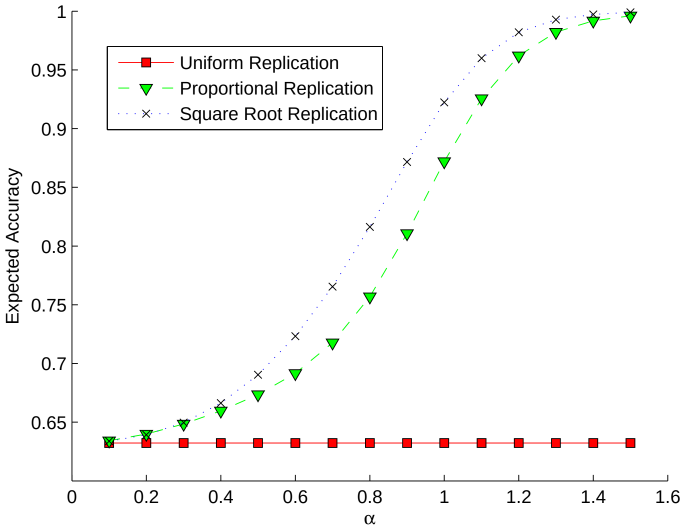


Fig. 1: Expected accuracy for retrieving the top-1 document, A_1, as a function of the power law exponent, α, for different replication strategies, when the number of nodes queried, $z = 1000$.

and for square root replication strategy, where $r_j = R\frac{\sqrt{q_j}}{\sum\sqrt{q_j}}$, the expected accuracy A_1^s is given by

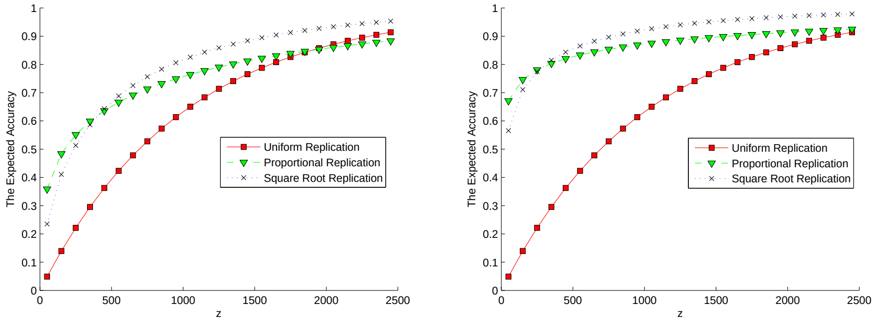
$$A_1^s = \sum q_j(1 - (1 - \rho\frac{\sqrt{q_j}}{\sum\sqrt{q_j}})^z) \tag{13}$$

To examine the effect of the different replication strategies, we considered a PAC configuration in which it is assumed that there are 1 million documents in the collection ($m = 10^6$), and 10,000 nodes in the network ($n = 10^4$). Each node is able to index 1000 documents ($\rho = 1000$).

Figure 1 shows the expected accuracy, A_1, when retrieving the top-1 document as a function of the power law exponent for different replication strategies. Here, we have assumed that the query distribution follows a power law, and have fixed the search size to 1000 nodes, ($z = 1000$). The square root replication strategy performs better than the proportional replication strategy, and grows more rapidly as α increases.

Figures 2a and 2b show the expected accuracy, A_1, when retrieving the top-1 document, as a function of the search size, z, for different replication strategies, and for $\alpha = 0.8$ and $\alpha = 1$ respectively. We observe that square root replication is inferior to proportional replication, when the search size is small. Note however, that as the search size increases, proportional replication improves more slowly, and square root replication performs better.

We can extend our analysis to the case where we are interested in the top-k retrieved documents, rather than only the top-1. Theoretically, an infinite number of queries can be issued by users, and many queries can retrieve the same documents. However, to simply our analysis of expected accuracy we assume a finite number of queries, $|Q|$, which is certainly true for a finite period of time. The top-k documents retrieved by each query, $\mathcal{D}_k(j)$, are likely to be non-disjoint, i.e. two queries might retrieve some documents in common. Thus, replication should be based on the distribution of retrieval frequency of the documents, rather than the query distribution directly.



(a) $\alpha = 0.8$

(b) $\alpha = 1.0$

Fig. 2: Expected accuracy for retrieving the top-1 document, A_1, as a function of the search size, z, for different replication strategies

To solve top-k retrieval problem, let us define a document retrieval frequency set $\mathcal{Q}'$ which holds the distribution of retrieval frequency of the documents in the collection. Thus, for each $q'_i \in \mathcal{Q}'$, we have

$$q'_i = \sum_{j=1}^{|\mathcal{Q}|} q_j \zeta(j, i)$$

where

$$\zeta(j, i) = \begin{cases} 1 \text{ if document } i \text{ is in query } j\text{'s top-}k \text{ result list.} \\ 0 \text{ otherwise.} \end{cases}$$

We can then transform Equation (10) to

$$A_k = \sum q'_i (1 - (1 - \frac{r_i}{n})^z) \tag{14}$$

where the replication rate for document i, r_i, is computed based on the corresponding q'_i. Since the expected accuracy is essentially a weighted mean, we can exploit the overlap of retrieved documents of queries and are able to simplify the top-k retrieval into an equation which is akin to the top-1 retrieval.

4 Communication Cost

In this Section we consider the communication cost associated with P2P search. We assume a probabilistic search architecture based on the PAC model. We make the following assumptions with regard to the system:

- a network size of $n = 1,000,000$ nodes.
  Several P2P services already exceed this number, e.g. Gnutella and BitTorrent, and the commercial P2P information retrieval system Faroo[1] currently claims 1 million users.
- a query rate of 1,000 queries a second.
  The estimated query rate of Google is 38,000 queries per second.[2] However, Google's query rate is based on a user community of about 150M unique users.[3] A query rate of 38,000 queries per second is equivalent to each of

[1] http://www.faroo.com/hp/p2p/p2p.html

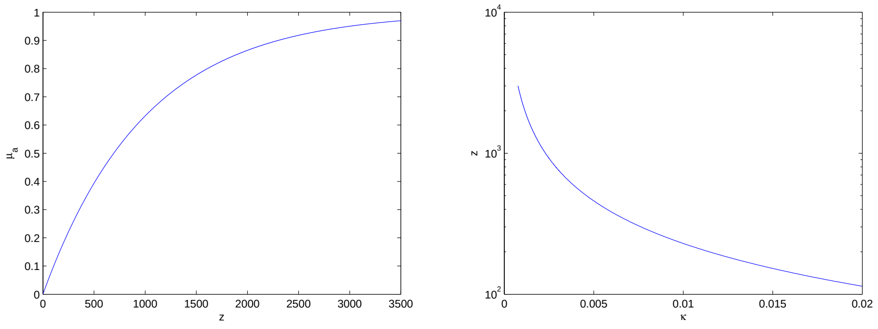
[2] http://searchengineland.com/by-the-numbers-twitter-vs-facebook-vs-googlebuzz-36709

[3] http://siteanalytics.compete.com/google.com+facebook.com+yahoo.com/

1 million nodes issuing over 2 queries a minute! A rate of 1000 queries per second corresponds to each node issuing almost 4 queries an hour, 24 hours per day, which would seem like an upper bound on any realistic query rate.

- a collection size of 10 billion documents to be indexed.
  Currently, it is estimated that Google indexes approximately 20 billion documents, while Bing and Yahoo index approximately 12 billion documents.[4]
- a required expected retrieval accuracy of 90%.
  If we are only interested in a single document, then the accuracy is given by Equation (9). Thus, we must choose a combination of the number of nodes the query is sent to, z, and the local storage capacity, ρ. Let $\kappa = \frac{\rho}{m}$, denote the fraction of the global collection indexed at a node.
- a minimum storage of 5 GB, available at each node and a maximum of 10 GB.
  This allows a node to index $\kappa = \frac{1}{1000}$ of the global document collection as discussed shortly.

For uniform replication, Figure 3a illustrates the expected accuracy as a function of the number of nodes queried when each node randomly samples $\kappa = \frac{1}{1000}$ of the global document collection. For 90% accuracy we need to query approximately 2,300 nodes. Figure 3b shows the number of nodes that need to be queried to obtain 90% accuracy as function of κ.



(a) Expected accuracy as a function of the number of nodes queried when $\kappa = 0.001$

(b) The number of nodes queried for 90% accuracy as a function of κ. Note that this is a log-linear plot.

Fig. 3: Relationship between number of nodes queried (z), the fraction of the global collection indexed at a node (κ), and the expected accuracy (μ_e) for uniform replication

In order to estimate the communications load we further assume:

- an average of 2 bytes per character.
  This is based on UTF-8 encoding, where each character takes between 1 - 4 bytes depending on the language used.[5]
- a query message size of 300 bytes.
  Analysis of query logs [11] has shown that the average query size is 2.5 terms or 30 characters. This corresponds to about 60 bytes per query message.

[4] http://www.worldwidewebsize.com

[5] http://tools.ietf.org/html/rfc3629

However, we must also assume some overhead associated with the underlying TCP/IPv6 protocol. We therefore conservatively assume a query message size of 300 bytes. Therefore if this message must be sent to $z = 1,000$ peers, the communication cost associated with *sending* a query is 300 KB.

Finally, in order to estimate the communication bandwidth needed to respond to a query, we assume the following:

- we are only interested in the top-10 documents.
  Analysis of commercial search engine query logs show that users rarely look beyond the top-10 documents. Thus, when a user issues a query, each node only needs to return its top-10 URLs. If, however, the user requests to see results 11-20, we could ask the same nodes to return their top 11-20, which would again be merged and re-ranked at the node originating the query.
- a query response size of 1KB.
  We estimate that each result (result name, hyper-link, snippet, minimal surrounding XML etc) requires no more than 400 characters or 800 bytes. Since the query result is entirely alphanumeric, it can usually be compressed to 10% of its original size. This is common practice with modern web servers.[6] Thus the total bandwidth required to answer a query is simply 800 bytes per result, multiplied by 10 results per query, times 0.1 compression factor, i.e. 800 bytes. We round this to 1 KB to account for TCP/IPv6 overheads.

4.1 Communications load for uniform replication

Based on the previous assumptions, we first consider the uniform replication strategy. From Equation (8), the expected accuracy of 90% can be obtained by sending the query to 2,300 nodes, assuming each node indexes 0.1% of the global collection.

We are now in a position to calculate the total communication load of such a system. For broadcasting the query and receiving the response from 2,300 peers, the total cost per query is approximately 3 MB. For 1,000 queries per second, the total traffic generated is 3 GB/s.

Note that this traffic is spread throughout the internet. The total internet traffic in 2009 was approximately 4,630 GB/s [14, 15] and is forecast to grow by 50% each year, primarily due to video traffic. Using the 2009 figures, the traffic generated by a PAC web IR service would only constitute 0.065% of the global internet traffic. Thus, web search using an unstructured P2P network will not impose a significant global communication cost.

As well as the global communication cost, it is useful to consider the requirements placed on each node, both in terms of storage and bandwidth.

We now estimate the resource requirements on each peer participating in the search. We have assumed that each node randomly samples 1/1000 of the 10 billion documents which need to indexed. This implies that each peer must index 10 million documents, which must first be crawled. The Akamai internet report[7] states that the global average internet connection speed is approximately 200 KB/s. In the developed nations it is considerably higher, but we do not account for this here. If we assume that 25% of this bandwidth can be utilized (say, during the peer's idle time), it will take approximately 58 days to complete the crawl, assuming that the average size of a document on the Web is 25KB[8].

[6] http://httpd.apache.org/docs/2.0/mod/mod deflate.html

[7] "Akamai report: The state of the internet, 3rd quarter, 2010", http://www.akamai.com/stateoftheinternet/

[8] http://www.optimizationweek.com/reviews/average-web-page/

The crawled documents, representing 250GB of data, can be indexed using approximately 10 GB of disk space which would record term frequencies and positions as well as other statistical measures. This is typical of popular information retrieval packages such as Lucene[9]. We are aware that some machines may not have 10GB of disk storage available for this service. However, lossless compression [16] can reduce the size of the index by utilizing efficient data structures, and lossy index compression techniques [17] have been shown to reduce the size of the index by 50 to 70% with minimal loss in precision.

Using efficient Trie structures, only small percentages of the index need to be read and loaded into memory, and the system can answer queries using no more than 500 MB of the peer's memory, as has been demonstrated by systems such as Lucene.

For a PAC web IR system of 1 million nodes answering 1,000 queries per second, each peer on average would have to answer 2.3 queries per second. The corresponding bandwidth needed is 0.69 KB/s in the download direction and 2.3 KB/s in the upload.

To summarize, each peer would need to contribute 5-10 GB of disk space, 500 MB of memory, and approximately 0.69 KB/s download as well as 2.3 KB/s upload from the peer's bandwidth for query answering as well as 50KB/s during idle time for crawling. Both the local communication, and disk and memory requirements appear reasonable.

Table 1: The number of nodes queried and the corresponding communication cost, for uniform, proportional and square root replication strategies, for $\alpha = 0.8$ and 1 to obtain an expected accuracy of 90%.

	$\alpha = 0.8$		$\alpha = 1.0$	
	Nodes Queried	Cost/Query (MB)	Nodes Queried	Cost/Query (MB)
Uniform Replication	2300	3.000	2300	3.000
Proportional Replication	2750	3.575	1180	1.534
Square Root Replication	1650	1.534	780	1.014

4.2 Communication load for non-uniform replications

As mentioned previously, the value of α ranges from 0.8 to 1.0. We can use Equations (12) and (13) to calculate the communication costs for an expected accuracy of 90% for non-uniform replications. The results are summarized in Table 1. We observe that for both values of α, the square root replication strategy needs to query fewer nodes than for the uniform distribution. For $\alpha = 0.8$ this reduces the communication bandwidth by about 50%, while for $\alpha = 1$, the bandwidth is reduced by about two thirds. It is interesting to note that for $\alpha = 0.8$, proportional replication performs worse than a uniform replication.

For a system servicing 1,000 queries per second, the communication costs correspond to between 0.03% and 0.07% of the global internet traffic.

[9] http://lucene.apache.org/

5 Conclusion

This paper investigated the feasibility, with respect to communication bandwidth, of performing web-scale search on an unstructured, distributed peer-to-peer network. The unstructured nature of the network necessitates that the search is probabilistic in nature. While this has been previously recognized, prior work has not considered the accuracy of the search, nor the probability of attaining said accuracy.

Communication cost is often cited as the limiting factor in the deployment and scalability of P2P search. For a uniform replication policy, that ignores the query distribution and the popularity of documents, it was shown that the communication load produced by the P2P system was only 0.07% of global internet traffic, in order to guarantee an expected accuracy of 90%. In addition, the local communication load placed on each peer is approximately 2.3KB/s in the upload direction and 0.69KB/s for download. Thus, the communication overhead is well below any level that would preclude P2P search.

The communication cost can be reduced by replicating documents based on their popularity. Two popular replication policies are proportional and square root, and we extended the theoretical analysis of expected accuracy for PAC search to these two non-uniform replication policies. For $\alpha = 0.8$, the proportional policy is actually worse than uniform, but is better than uniform when $\alpha = 1$. The square root policy is superior to uniform for both values of α. However, we note that the square root policy is not optimum for maximizing accuracy. An optimum replication policy is left for future work.

In Section 4.1 we found that a node would take approximately 58 days to complete each iteration of a web crawl. This could degrade the freshness of results and decrease the relevance of documents. A solution to this issue, could be the use of cloud computing resources such as Amazon EC2. These servers, financed perhaps with small individual donations made to a non-profit organization, can be scaled up or down based on available funding, and can be used to continuously crawl and index the web. The nodes in the P2P network could then refresh their index in fragments continuously using BitTorrent from these cloud computing based index servers. Apart from drastically reducing the number of days required for a crawl, this would have an added advantage of reducing the peer's workload of crawling and indexing.

Of course, communication cost is not the only factor that might prevent wide scale P2P web search. Latency, i.e. the time to respond to a query, is also a factor. Latency is usually considered to be proportional to the number of nodes queried. However, we note that for an unstructured P2P search architecture, we do not have to wait for all peers to respond before displaying a *partial* result list. The heterogeneity of peers in any P2P network makes this an inevitability. The analysis, modelling, and minimization of latency in web search within an unstructured P2P architecture is part of our future research.

Acknowledgment

We are thankful for the comments and suggestions provided by the referees, which has considerably helped in improving this paper. Ruoxun Fu gratefully acknowledges the support of BT.

References

1. J. Li, B. T. Loo, J. Hellerstein, F. Kaashoek, D. R. Karger, and R. Morris, "On the Feasibility of Peer-to-Peer Web Indexing and Search," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

2. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.* ACM, 2003, pp. 407–418.
3. E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," *Peer-to-Peer Systems*, pp. 261–269, 2002.
4. M. Zhong, J. Moore, K. Shen, and A. L. Murphy, "An evaluation and comparison of current peer-to-peer full-text keyword search techniques," in *Proc. of the International Workshop on the Web Databases (WEBDB)*, 2005.
5. Y. Yang, R. Dunlap, M. Rexroad, and B. F. Cooper, "Performance of Full Text Search in Structured and Unstructured Peer-to-Peer Systems," in *Proceedings of the 25th IEEE Internation Conference on Computer Communications, 2006, Barcelona, Spain.* IEEE, 2006.
6. B. F. Cooper, "An optimal overlay topology for routing peer-to-peer searches," in *Proceedings of the 6th International ACM/IFIP/USENIX Middleware Conference, 2005, Grenoble, France*, 2005.
7. W. W. Terpstra, J. Kangasharju, C. Leng, and A. P. Buchmann, "Bubblestorm: resilient, probabilistic, and exhaustive peer-to-peer search," in *SIGCOMM*, 2007, pp. 49–60.
8. D. Malkhi, M. Reiter, and R. Wright, "Probabilistic quorum systems," in *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing.* ACM, 1997, p. 273.
9. R. Ferreira, M. Ramanathan, A. Awan, A. Grama, and S. Jagannathan, "Search with probabilistic guarantees in unstructured peer-to-peer networks," 2005.
10. E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications.* ACM, 2002, pp. 177–190.
11. R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, and F. Silvestri, "The impact of caching on search engines," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, July 23-27, 2007, Amsterdam, The Netherlands.* ACM, 2007.
12. I. J. Cox, R. Fu, and L. K. Hansen, "Probably approximately correct search," in *Proc. of the International Conference on Theoretical Information Retrieval (ICTIR)*, 2009.
13. I. J. Cox, J. Zhu, R. Fu, and L. K. Hansen, "Improving query correctness using centralized probably approximately correct (pac) search," in *ECIR*, 2010, pp. 265–280.
14. C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM, August 30-September 03, 2010, New Delhi, India.* ACM, 2010.
15. "http://dtc.umn.edu/mints/."
16. I. J. Witten, A. Moffat, and T. C. Bell, *Managing Gigabvtes: Compressing and Indexing Documents and Images*, 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1999.
17. E. S. de Moura, C. F. dos Santos, D. R. Fernandes, A. S. Silva, P. Calado, and M. A. Nascimento, "Improving Web search efficiency via a locality based static pruning method," in *Proceedings of the 14th international conference on World Wide Web, May 10-14, 2005, Chiba, Japan*, 2005.