Scalability Analysis of Distributed Search in Large Peer-to-peer Networks

Weimao Ke
College of Computing and Informatics
Drexel University, Philadelphia, PA 19104
Email: wk@drexel.edu, http://lincs.ischool.drexel.edu

Javed Mostafa
School of Information and Library Science
University of North Carolina, Chapel Hill, NC 27599
Email: jm@unc.edu, http://lair.unc.edu

***Abstract*—We study decentralized searches in large-scale, self-organized peer-to-peer networks and investigate the influences of network size and degree distribution (neighborhood size) on search efficiency. Experimental results show that searches are efficient and scalable in large networks, especially with large neighborhood sizes (degrees). Analysis of the data supports a proposed scalability model, in which search path length L (efficiency) is proportional to a poly-logarithmic function of network size N, with degree d_m (majority neighborhood size) as the log base. The model explains 90% (R^2) of variances in search path lengths. Search time (search path length) predicted by the model shows great potential for efficient searches in real-scale networks of up to a billion distributed systems.**

***Keywords*-Distributed search; Scalability; Efficiency; Peer-to-peer networks; Decentralization**

I. Introduction

Decentralization is the nature of many naturally, socially, and technologically grown structures that scale. The Web and the Internet operate in a rather decentralized manner without explicit global control. On the Internet, fundamental technologies such as routing and lookup operations are decentralized by design and are able to scale with the rapid growth of the network. According to Vint Cerf, the decentralized nature of the Internet is a feature of its resilience and scalability [4].

From the perspective of the Internet of everything, where any tiny gadgets and daily items can be attached to a highly interconnected digital world, the question of finding relevant pieces of information about and on these devices is staggeringly challenging. While centralization is likely to fail in the long term, decentralization represents the future of technological innovation; searching is an essential part of the trend.

Our research envisions a fully decentralized architecture in which individual search engines can interconnect and contribute to the collective power of finding relevant information, through distributed routing or network traversal. The goal of this research is to understand the general mechanisms by which a large number of distributed systems can work together to support scalable search and retrieval operations. It aims to explore alternative search engine architectures that can function, scale, and cope with the increasing magnitude and dynamics of networks such as the Web.

II. Related Work

Related challenges facing distributed or decentralized searches have been studied in areas of distributed (federated) information retrieval, peer-to-peer networks, multi-agent systems, and complex networks [3], [13]. Classic distributed IR research has focused on distributed database content and characteristics discovery, database selection, and result fusion in a relatively small number of distributed, persistent information collections [7], [19], [15], [18].

Peer-to-peer IR research often involves a larger number of distributed systems which dynamically join and leave the community. Related projects have employed techniques such as distributed hashing tables (DHTs) in structured P2P networks and semantic overlay networks (SONs) in unstructured networks for efficient discovery [3], [20], [5]. Agent-based modeling has proven to be a powerful tool in distributed information retrieval (IR) research [8]. The agent paradigm has been extensively used to model processes such as P2P search [21], intelligent crawling [16], and expert finding [12].

The central idea of peer segmentation or clustering in frameworks such as SONs to support efficient decentralized searches has also been studied in complex network research. In networks with small world properties, studies have demonstrated that globally relevant targets can be found efficiently through collaboration of distributed, local intelligence in large networks [17], [14], [2].

Our research has studied related decentralized/distributed information retrieval problems in light of network formation and clustering, emergent from interconnectivity among distributed systems. In a series of large-scale IR experiments we conducted, network clustering based on semantic overlay was found to be useful for decentralized searches, however, with qualifications [9], [10]. The best search efficiency and effectiveness were supported by a very specific level of network clustering. Any departure from that fine-tuned level, i.e. stronger or weaker clustering, degraded search performance significantly [11]. So far we have investigated related networks of various sizes using multi-agent simulation. Results from these studies showed very promising search performances. However, further investigation is needed to understand other network structure variables' impact on search scalability.

III. Focus on Search Scalability

In this research, we simulate classic IR operations in a distributed setting and study search performances with regard to the following research aspects: 1) the impact of network size N on search efficiency , 2) the impact of agent (local) neighborhood size N_r on search efficiency in various sizes of networks, and 3) projection of decentralized search performance (efficiency in particular) when the network continues to grow.

A. Proposed Scalability Model

This study is focused on search efficiency and scalability with growing network sizes N and varied (distributed system) neighborhood sizes d (degree distributions) with optimal network clustering. We expect to validate a scalability model which will enable us to project search efficiency in web-scale networks. We discuss the model below.

Let L denote search path length, i.e. the number of hops (distributed systems) a search query traverses the network to reach a desired target (with relevant information). According to [13] and several studies in distributed IR, when network clustering is optimal, a reasonable relation between $\hat{L}$ (expected value of L based on relevance/similarity searches) and network size N (the number of distributed systems in the network) is:

$$\hat{L} = \beta' \cdot (\log_b N)^\lambda \tag{1}$$

where β' is a constant and b is the logarithmic base. λ is an exponent parameter to be identified with empirical data.

Assume the majority of distributed systems (hops) have a neighborhood size (number of interconnected systems) d_m. Let L_g denotes the ideal search path length given a (imagined) perfect, global index of all distributed systems. For example, when degree $d_m = 2$, L_g can be seen as the number of steps needed to perform a binary search (traversal of a binary tree) on N nodes. With a sorted binary tree ($d_m = 2$), it is known that the number of nodes/branches to be traversed in a search among N nodes is $\log_2(N)$. Likewise, with a tree in which each nodes has d_m branches, the ideal search path length L_g can be computed by:

$$L_g \quad = \quad \log_{d_m} N \tag{2}$$

Without a global index discussed above, searches will become less efficient. In this case we expect a longer search path length in a distributed network. We reason that the expected search path length $\hat{L}$ is greater than and is no longer a linear function to $\log_{d_m} N$. In light of Equation 1, it should be a poly-logarithmic function of N or a power function of $\log_{d_m} N$ (i.e. using d_m as the logarithmic base). Hence, Equation 1 becomes:

$$\begin{aligned} \hat{L} &= \beta \cdot (\log_{d_m} N)^\lambda && (3) \\ &= \beta \cdot (\log N / \log d_m)^\lambda && (4) \end{aligned}$$

where β is a constant and d_m the neighborhood size (degree) of majority distributed systems. To simulate real networks, a power-law function will be used for degree distribution $d \in [d_m, d_x]$, where d_m is the min degree (which the vast majority have in a power law) and d_x is the maximum value (which only a small number of nodes have).

In this study, we expect to validate the above scalability model with large-scale experiments, identify the exponent λ, estimate the β coefficient with varied N and d_m settings, and predict potential search efficiency in real-scale environments with millions to billions of distributed systems.

IV. Simulation Framework and Algorithms

We have developed a decentralized search simulation framework based on multi-agent systems for finding relevant information in distributed settings. Each agent represents an IR system, which has its document collection and can connect to others to route queries. The simulation system was implemented in Java the multi-agent system JADE [1] and full-text search library Lucene [6].

In the simulation framework, each agent builds an index on a local document collection and connects to a number of neighbors (a variable in this study) for help with unanswered queries. When an agent receives a query/task, it first searches its local collection and, if the result is unsatisfactory, forwards the query to one of its neighbors most likely to have relevant information. The query routing continues until relevant results have been found or when it reaches a limit (i.e. max search path length). Further details on the simulation framework can be found in [11].

The subsections below elaborate on specific algorithms implemented in the framework for 1) information representation and weighting (to represent documents and queries), 2) neighbor (agent) representation, 3) neighbor selection (search) method, and 4) a network interconnectivity (clustering) function.

A. Basic Functions

*1) TF*IDF Information Representation:* Each agent processes information it individually has and produces a local term space, which is used to represent each information item using the classic TF*IDF (Term Frequency * Inverse Document Frequency) weighting scheme. Note that IDF values are based on the agent's local collection.

*2) DF*INF Neighbor Representation:* An agent uses a meta-document to represent each of its neighbors. The weight of term t in a meta-document is computed by: $W'(t) = df'(t) \cdot log(\frac{N'_b}{nf'(t)})$, where $df'(t)$ is the number of documents in the neighbor agent (collection) containing term t, N'_b is the total number of the agent's neighbors (meta-documents), and $nf'(t)$ is the number of neighbors containing the term t. We refer to this function as *DF*INF*, or Document Frequency * Inverse Neighbor Frequency.

3) Similarity Scoring Function: Given a query q, the similarity score of a document d matching the query is computed by: $\sum_{t \in q} W(t) \cdot coord(q,d) \cdot queryNorm(q)$, where $W(t)$ is the weight of term t given by the above TF*IDF or DF*INF, $coord(q,d)$ a coordination factor based on the number of terms shared by q and d, and $queryNorm(q)$ a normalization value for query q given the sum of squared weights of query terms. This scoring function is used to compute query-document similarities as well as query-metadocument (query-neighbor) similarities.

B. Neighbor Selection (Search) Methods

We use the following strategies to decide which neighbors should be contacted for the query: 1) Random Walk (RW), 2) SIM Search which selects the neighbor with the highest similarity score, 3) DEG Search which selects the one with the highest degree, and 4) Sim*Deg which combines similarity and degree scores to determine the best neighbor.

C. Interconnectivity and Network Clustering

To interconnect agents, the first step is to determine how many links (degree d_u) each distributed agent/system u should have. Once the degree is determined, the system will interact with a large number of other systems (from a random pool) and select only d_u systems as neighbors based on a connectivity probability function guided by the clustering exponent α. Based on the ClueWeb data, given the number of incoming hyperlinks d'_u of system/site u, the normalized degree is computed by:

$$d_u = d_m + \frac{(d_x - d_m) \cdot (d'_u - d'_m)}{d'_x - d'_m} \tag{5}$$

where d'_x is the maximum degree value in the hyperlink in-degree distribution and d'_m the minimum value in the same distribution. Once degree d_u is determined from the degree distribution, a number of random systems/agents will be added to its neighborhood pool such that the total number of neighbors $\hat{d}_u \gg d_u$ ($\hat{d}_u = 1,000$ in this study). Then, the agent in question (u) queries each of the $\hat{d}_u$ neighbors (v) to determine their topical distance r_{uv}. Finally, the following connection probability function is used by system u to decide who should remain as neighbors (to build the interconnectivity overlay):

$$p_{uv} \propto r_{uv}^{-\alpha} \tag{6}$$

where α is the *clustering exponent* and r_{uv} the pairwise topical (search) distance. The finalized neighborhood size will be the expected number of neighbors, i.e., d_u. With a positive α value, the larger the topical distance, the less likely two systems/agents will connect. Large α values lead to highly clustered networks while small values produce random networks with many topically remote connections.

V. Experimental Design

A. Data Collection

We use the ClueWeb09 Category B collection in experiments, which contains 428,136,613 nodes (pages) and 454,075,604 edges (hyper-links). Additional details about the ClueWeb09 collection can be found at http://boston.lti.cs.cmu.edu/Data/clueweb09/. Using the ClueWeb09 collection, we treat a web site/domain as a distributed system/agent and use hyperlinks between sites to construct the initial network.

B. Search Task - Rare Item Search

We rely on existing evidence in data to do automatic relevance judgment. We use documents (with title and content) as queries for decentralized searches. From the first 1000 web domains constructed above, we select as queries 12 random web pages with at least 3 in-links. The final set of query documents include (all trecids with prefix *clueweb09-en000*): 1-42-03978, 1-73-04287, 1-90-26216, 2-73-04700, 2-91-14776, 3-27-30577, 3-30-28328, 3-51-10345, 3-55-31539, 4-61-19060, 4-72-24215, 4-92-04942. The search task is to find the exact copy of a given document (query) distributed in the network.

C. Variables

1) Network Model and Sizes: We first construct a list of all web domains in the category B subset with at least one in-link in the provided web graph. We take the first 1000 web domains/sites to construct the 1000-system network and extend it to 2000, 5000, 10^4, and 10^5 systems in additional experiments. Network clustering is performed using the method described in Section IV-C to establish individual system neighborhoods. We use clustering exponent $\alpha = 2$ in experiments.

2) Degree Distribution: $[d_m, d_x]$: We use various degree ranges $d_u \in [4, 8]$, $[16, 64]$, $[64, 128]$, and $[256, 512]$, to examine the impact of degree distribution (neighborhood size) on decentralized searches. Note that with a power-law distribution, the min degree d_m which defines the lower-bound of the range is actually the degree value (neighborhood size) that the majority of nodes/systems have.

3) Maximum Search Path Length L_{max}: The maximum search path length L_{max} specifies the longest search path (TTL) allowed for query traversal. With our focus on search efficiency/scalability, we set L_{max} as the total number of systems N in experiments to achieve best possible effectiveness.

4) Parameter Settings: We use full combinations of the following variables/parameters in experiments, i.e., 5 (network sizes $N \in [1000, 2000, 5000, 10^4, 10^5]$) × 4 (degree ranges $d \in [4, 8]$, $[16, 64]$, $[64, 128]$, $[256, 512]$) × 4 (search methods RW, SIM, DEG, and SimDeg).

D. Evaluation

To evaluate effectiveness, we use the classic metric F_1 based on precision and recall: $F_1 = 2PR/(P + R)$, averaged for all queries. For efficiency, we use the average search path length L of all queries. For scalability analysis, we run experiments on different network size scales. Search path length L as a function to network size N and neighborhood size d_m (proposed in Equation 4) will be modeled and analyzed with experimental data.

VI. Results

Experimental results show that searches were efficient and scalable in large networks, especially with large neighborhood sizes (degrees). Analysis of the data supports the proposed scalability model (Equation 4), in which search path length (efficiency) is proportional to $(\log_{d_m} N)^{\lambda}$, where d_m is the majority degree value (neighborhood size) and N is the number of distributed systems in the network. The model explains 90% (R^2) of variances in search path lengths. Efficiency (search path length) values predicted by the model show great promise for search scalability in real-scale networks, e.g., of a billion distributed systems. We discuss detailed results below.

A. Influences of Neighborhood Size

Figures 1 (a), (b), (c), and (d) show effectiveness of the experimented search methods in the 1000-, 2000-, $10,000$-, and $100,000$-system networks respectively. Each sub-figure plots effectiveness (F_1) vs. neighborhood size $d_m \in [4, 16, 64, 256]$. In most experimental settings, SIM, DEG, and SIM*DEG methods achieved perfect effectiveness scores $F_1 = 1.0$ and greatly outperformed the RW baseline. Overall, effectiveness improved when larger d_m degree values. The superior effectiveness was due to the fact that maximum search path length (TTL) was set to be the total number of distributed systems in the network, allowing searches to traverse the network thoroughly.

Figures 2 (a), (b), (c), and (d) show the search methods' efficiency in terms of search path lengths. Each sub-figure plots efficiency (search path length) vs. neighborhood size d_m (on log/log coordinates). The RW baseline stays roughly constant for different d_m values because it was not selective in the search process even when there were more neighbors to collaborate with. SIM, DEG, and SimDeg methods all achieved much better efficiency with larger neighborhoods/degrees. For example, when degree d_m increased from 16 to 256 (a 16-time increase), SIM search efficiency improved from more than $10,000$ hops to roughly 200 hops in the $100,000$-system network (a 50-time improvement).

B. Influences of Network Size

Figures 3 and 4 present the effectiveness and efficiency results from a different perspective. Each sub-figure in Figure 3 plots effectiveness F_1 vs. network size N for a given neighborhood size (degree) d_m = 4, 16, 64, or 256. In sub-figures (c) and (d), SIM, DEG, and SimDeg achieved perfect F_1 scores with large degree d_m values (64 and 256). RW baseline effectiveness was much worse.

Figure 4 sub-figures plot efficiency (search path length) vs. network size N. In these sub-figures, when network size N increased, search efficiency degraded (with larger search path lengths). SIM (as well as DEG and SimDeg) consistently outperformed the RW baseline in all settings. The proportional performance difference was much greater with larger neighborhood size d_m (e.g., compare $d_m = 256$ and $d_m = 4$ sub-figures). In the $100,000$-system network with $d_m = 256$ (in sub-figure (d)), for example, SIM search path length was about 200 whereas the search path length for RW baseline was around $80,000$ (a 400-time difference).

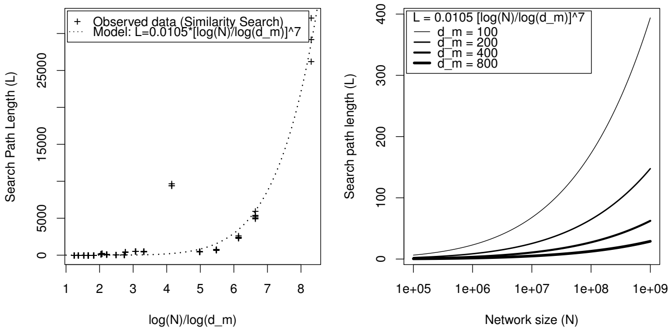
C. Scalability of Searches

With the proposed scalability function in Equation 4, we relied on a generalized regression model to analyze data, validate the model, and estimate related parameters. Using $\hat{L} \sim \beta \cdot (\log N / \log d_m)^{\lambda}$, we modeled experimental results of SIM searches with different exponents $\lambda \in [2..10]$ in the BoxCox process. The best fit was achived by $\lambda = 7$, with model estimates in Table I:

<table>
<tr><td colspan="5">Model: $L \sim 0 + \beta(\log_{d_m} N)^7$</td></tr>
<tr><td></td><td>Coef Estimate</td><td>Std Error</td><td>t value</td><td>$Pr(>|t|)$</td></tr>
<tr><td>β</td><td>0.0105</td><td>0.000365</td><td>28.9</td><td>$5.8E^{-47}$ ***</td></tr>
<tr><td colspan="5">$R^2 = 0.903$ (adj. 0.902), $F = 833$ on 1 and 89 DF</td></tr>
</table>

TABLE I
SIM Search: Search Path length vs. Network size. L is search path length; N is network size; d_m is the (min) neighborhood size (number of neighbors).

As shown in Table I, the resulted model $\hat{L} = 0.0105 \cdot (\log_{d_m} N)^7$ has a very large coefficient of determination: $R^2 = 0.903$. Figure 5 shows observed data of SIM searches in experiments and expected data using estimates in Table I. Except for a couple of outliers, the model fits observed data very well.

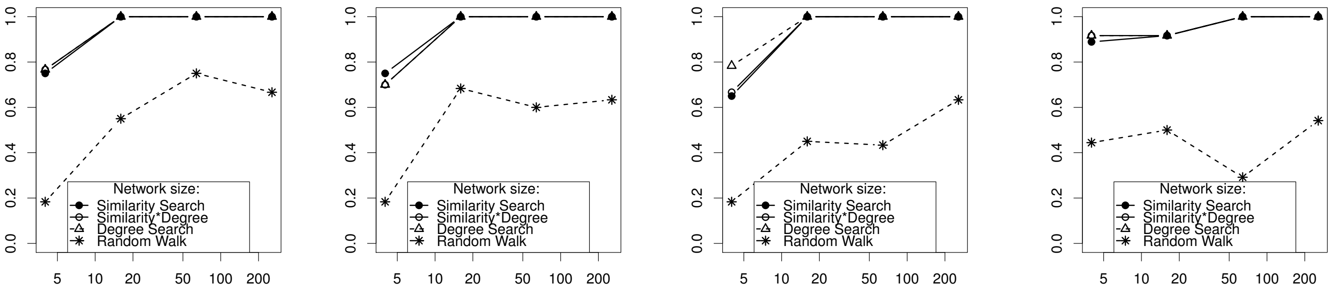


(a) Experimented data (b) Predicted data

Fig. 5. (a) Experimented scalability of SIM Search: Search path length vs. $\log_{d_m} N$, where N is network size and d_m is (min) neighborhood size. (b) Predicted scalability of SIM Search in real-scale networks of up to a billion systems ($N \in [10^5..10^9]$). X denotes network size (log-transformed); Y is predicted search path length for SIM search.

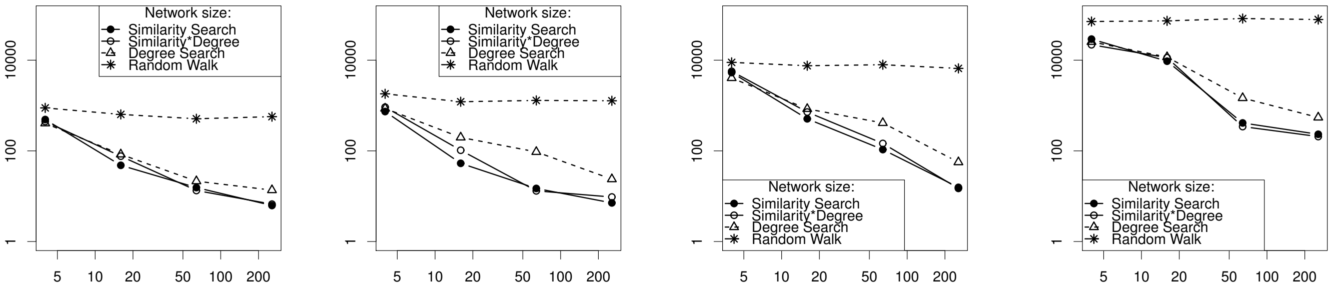
D. Model Prediction and Implications

Overall the scalability analysis supports search path length L as a poly-logarithmic function to network size N with neighborhood size (degree) d_m as the log base – that is $\hat{L}$ can be estimated by $\beta(\log_{d_m} N)^{\lambda}$.



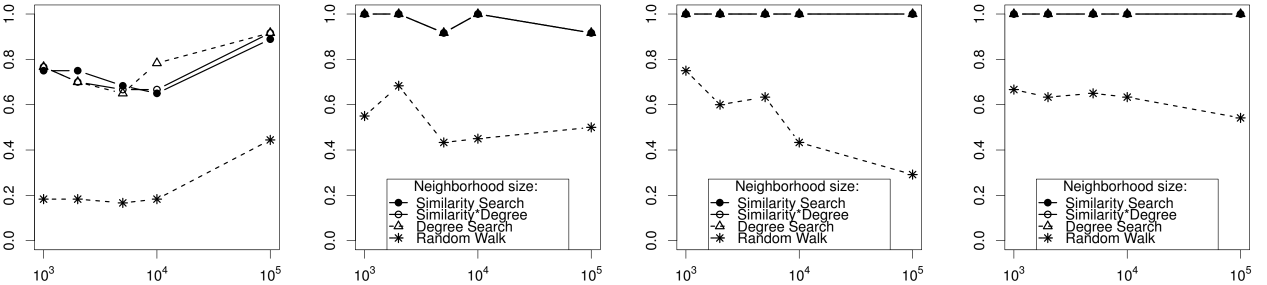
a) Network size $N = 1,000$ b) Network size $N = 2,000$ c) Network size $N = 10,000$ d) Network size $N = 100,000$

Fig. 1. Effectiveness vs. neighborhood size with varied network sizes. X denotes (min) neighborhood sizes $d_m \in [4, 16, 64, 256]$. Y denotes search effectiveness F_1. X is log-transformed.



a) Network size $N = 1,000$ b) Network size $N = 2,000$ c) Network size $N = 10,000$ d) Network size $N = 100,000$

Fig. 2. Efficiency vs. neighborhood size with varied network sizes. X denotes (min) neighborhood sizes $d_m \in [4, 16, 64, 256]$. Y denotes search path length L (efficiency). Both X and Y are log-transformed.



a) Neighborhood $d_m = 4$ b) Neighborhood $d_m = 16$ c) Neighborhood $d_m = 64$ d) Neighborhood $d_m = 256$

Fig. 3. Effectiveness vs. network sizes with varied neighborhood sizes. X denotes network size $N \in [1000, 2000, 5000, 10000, 100000]$. Y denotes search effectiveness F_1. X is log-transformed.

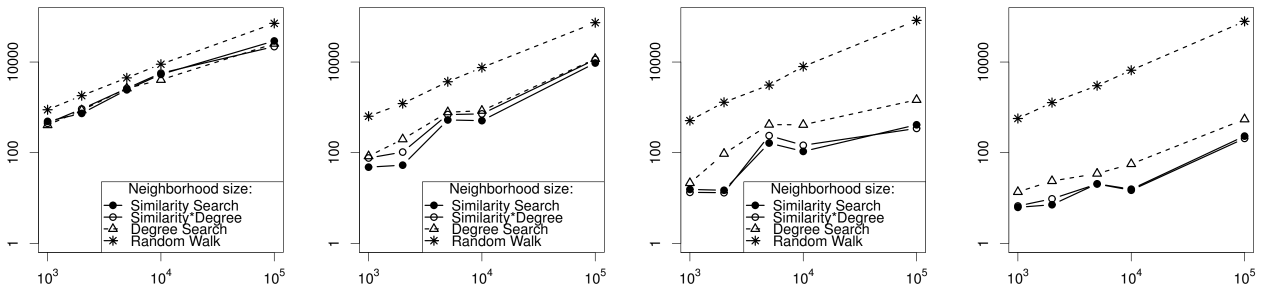
The model $L = 0.0105(\log_{d_m} N)^7$ based on coefficients in Table I can help estimate search efficiency (in terms of search path lengths) in real-scale networks with an even greater number of distributed systems. Given the model, Figure 5 (b) plots predicted data of search path length L vs. network size N for various degrees (neighborhood sizes) $d_m \in [100, 200, 400, 800]$.

As shown in Figure 5 (b), with degree $d_m = 400$ in a million-system network and $d_m = 800$ in a billion-system network, it is predicted to take less than 100 hops to locate a desired target. This efficiency and scalability is achieved by distributed systems that self-organize into a searchable network without global control.

Due to variances in data and model inaccuracy, actual search path lengths may vary significantly. Nonetheless, numbers predicted by the model point to the potential level of search efficiency/scalability. They show a great potential for fully distributed/decentralized searches to scale in growing information networks. In the future, we plan to conduct experiments on even larger-scale networks (such as networks of a million and more agents) and to verify the scalability prediction in this study.

VII. Conclusion

We studied decentralized searches in large-scale, self-organized information networks and investigated the influences of network size and degree distribution (neighborhood size) on search efficiency. Experimental results show that searches were efficient and scalable in large networks, especially with large neighborhood sizes (degrees). Analysis of the data supports the proposed scalability model, in which search path length (efficiency) is proportional to $(\log_{d_m} N)^\lambda$, where d_m is the



a) Neighborhood $d_m = 4$ b) Neighborhood $d_m = 16$ c) Neighborhood $d_m = 64$ d) Neighborhood $d_m = 256$

Fig. 4. Efficiency vs. network sizes with varied neighborhood sizes. X denotes network size $N \in [1000, 2000, 5000, 10000, 100000]$. Y denotes search path length L (efficiency). Both X and Y are log-transformed.

majority degree value (neighborhood size) and N is the number of distributed systems in the network. The model explains 90% (R^2) of variances in search path lengths. Efficiency (search path length) values predicted by the model show the great potential of search scalability in real-scale networks without central control or global knowledge.

Given the nature of decentralization and increasing challenges of big-data in the digital world, the future of information technology lies in decentralized methods that can adapt and scale. In this vision, a decentralized information retrieval architecture will over time provide better search results and scale more gracefully than today's "monolithic" search engines employed on the Web. The presented scalability model and related results offer important insight into how such a decentralized search architecture can scale with network growth. In future research, we plan to scale up our experiments to one million agents and to verify the efficiency projection presented in this paper based on the million scale. We will also investigate issues such as network congestion and failures in the distributed search setting.

Acknowledgment

This work was supported in part by the National Science Foundation under grant no. 1646955.

References

[1] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.

[2] M. Boguñá, D. Krioukov, and K. C. Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74 –80, 2009.

[3] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. In *Agents and Peer-to-Peer Computing*, pages 1–13, 2005.

[4] T. DAWES. We dont need a backup plan for the internet, says the guy who invented it, March 2013. [Online; posted 22-March-2013].

[5] C. Doulkeridis, K. Norvag, and M. Vazirgiannis. Peer-to-peer similarity search over widely distributed document collections. In *LSDS-IR '08: Proceeding of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval*, pages 35–42, New York, NY, USA, 2008. ACM.

[6] E. Hatcher, O. Gospodnetić, , and M. McCandless. *Lucene in Action*. Manning Publications, second edition edition, March 2010.

[7] D. Hawking and P. Thomas. Server selection methods in hybrid portal search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, New York, NY, USA, 2005. ACM.

[8] M. N. Huhns, M. P. Singh, M. H. Burstein, K. S. Decker, E. H. Durfee, T. W. Finin, L. Gasser, H. J. Goradia, N. R. Jennings, K. Lakkaraju, H. Nakashima, H. V. D. Parunak, J. S. Rosenschein, A. Ruvinsky, G. Sukthankar, S. Swarup, K. P. Sycara, M. Tambe, T. Wagner, and R. L. Z. Gutierrez. Research directions for service-oriented multiagent systems. *IEEE Internet Computing*, 9(6):65–70, November–December 2005.

[9] W. Ke and J. Mostafa. Strong ties vs. weak ties: Studying the clustering paradox for decentralized search. In *Proceedings of the 7th Workshop on Large-Scale Distributed Systems for Information Retrieval, co-located with ACM SIGIR 2009*, pages 49–56, Boston, USA, July 23 2009.

[10] W. Ke and J. Mostafa. Scalability of findability: effective and efficient ir operations in large information networks. In *SIGIR'10: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–81, 2010.

[11] W. Ke and J. Mostafa. Studying the clustering paradox and scalability of search in highly distributed environments. *ACM Trans. Inf. Syst.*, 31(2):8:1–8:36, May 2013.

[12] W. Ke, J. Mostafa, and Y. Fu. Collaborative classifier agents: studying the impact of learning in distributed document classification. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 428–437, New York, NY, USA, 2007. ACM.

[13] J. Kleinberg. Complex networks and decentralized search algorithms. In *In Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

[14] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623–11628, 2005.

[15] D. Lillis, F. Toolan, R. Collier, and J. Dunnion. Probfuse: a probabilistic approach to data fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146, New York, NY, USA, 2006. ACM.

[16] F. Menczer, G. Pant, and P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology*, 4(4):378–419, 2004.

[17] S. Milgram. Small-world problem. *Psychology Today*, 1(1):61–67, 1967.

[18] M. Shokouhi and J. Zobel. Federated text retrieval from uncooperative overlapped collections. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 495–502, New York, NY, USA, 2007. ACM.

[19] L. Si and J. Callan. Modeling search engine effectiveness for federated search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90, New York, NY, USA, 2005. ACM.

[20] G. Skobeltsyn, T. Luu, I. P. Zarko, M. Rajman, and K. Aberer. Web text retrieval with a p2p query-driven index. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 679–686, New York, NY, USA, 2007. ACM.

[21] H. Zhang and V. Lesser. A reinforcement learning based distributed search algorithm for hierarchical peer-to-peer information retrieval systems. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.