Towards a Decentralized Search Architecture for the Web and P2P Systems

Jie Wu
Swiss Federal Institute of Technology, Lausanne
School of Computer and Communication Sciences
1015 Lausanne, Switzerland
jie.wu@epfl.ch

Abstract

Search engines are among the most important applications or services on the web. Most existing successful search engines use a centralized architecture and global ranking algorithms to generate the ranking of documents crawled in their databases, for example, Google's PageRank. However, global ranking of documents has two potential problems: high computation cost, and potentially poor rankings. Both of the problems are related to the centralized computation paradigm. We propose a decentralized architecture to solve the problem in a P2P fashion. We identify three sub-problems in the big picture: a logical framework for ranking computation, an efficient way of computing dynamic local ranking, and a cooperative approach that bridges distributed local rankings and collective global ranking. In the paper we summarize the current knowledge and existing solutions for distributed IR systems, and present our new ideas. We also provide initial results, demonstrating that the use of such an architecture can ameliorate the above-mentioned problems for Web and P2P search engines.[1]

Keywords:

search engines, information retrieval, P2P systems, link analysis, swarm intelligence, decentralized algorithms

1. Introduction

Search engines for large scale distributed systems, e.g. the Web, the emerging P2P systems, face two radical challenges: a huge collection of documents and the processing of them in preparation for information retrieval, and the generation of a proper ranking of the huge number of documents. The state-of-the-art technologies of dealing with these two problems have big limitations such as high computation cost, potentially poor rankings, etc.. The focus of my PhD thesis work is to develop a decentralized architecture for efficiently searching and ranking documents with returned results of high quality. The work covers three main issues in the big picture of my new decentralized search architecture: firstly, a mechanism inspired by Swarm Intelligence of obtaining more dynamic and more semantically meaningful rankings of documents local to Web sites; secondly, a ranking algebra which provides the algebraic ground of computing document rankings; and finally, the idea of global Web site ranking which is the key to establish the global Web document ranking in a decentralized way, and a decentralized algorithm of computing the global Web site ranking. Substantial results have been achieved and further work is going on smoothly.

2. The Research Question

We brief the established IR models [13] here at first. Then we see why these models do not fit well the Web IR systems.

2.1 Centralized Search Systems

The classical model for a centralized IR system is: $S = (T, D, Q, \delta)$ where D is a document collection, Q is the set of queries, and $\delta : Q \rightarrow 2^D$ is the set of mappings which assign every query to a set of relevant documents. T is a set of distinct terms where two relations are defined: *synonymous*: $\rho \subset T \times T$ where $\rho(t_1, t_2)$ implies that t_1 is a synonym of t_2; *general* : $\gamma \subset T \times T$ where $\gamma(t_1, t_2)$ implies that t_1 is a more general term than t_2. Many IR systems use a thesaurus T to expand a user query by including synonyms of the keywords in the query. An example of a valid generalization is $\gamma(animal, fish)$. A partial ordering of documents can be defined based on the concept of generalization. Let $t(d_i)$ indicate the list of unique, non-mutual synonymous keywords[2] of document d_i. Partial ordering $\preceq$ is defined as:

$$t(d_1) \preceq t(d_2) \Leftrightarrow (\forall t' \in t(d_1))(\exists t'' \in t(d_2))(\gamma(t', t''))$$

This is a partial ordering because two documents with terms that have no relationship between any pairs of terms will be unordered. What is mainly used in query processing of IR systems is the so-called *inclusiveness* property of this model. An IR system is *inclusive* only when the documents corresponding to a general query q_1 must be a superset of all documents corresponding to a more specific query q_2 where $q_1 \preceq q_2$:

$$(q_1 \preceq q_2) \rightarrow (\delta(q_1) \supset \delta(q_2)) \qquad (q_1, q_2 \in Q)$$

The advantage of being *inclusive* is that, if two queries q_1 and q_2 are presented such that $\gamma(q_1, q_2)$, it is not necessary to retrieve from the entire document collection D for each query. Rather the system can obtain the answer set $\delta(q_1)$ for q_1, and then simply search $\delta(q_1)$ to obtain the $\delta(q_2)$.

2.2 Model of Distributed Information Retrieval

A model of decentralized IR can be built by partitioning the centralized IR system $S = (T, D, Q, \delta)$ into n local IR systems $S_i = (T_i, D_i, Q_i, \delta_i), i = 1, \cdots, n$, where T_i, D_i, Q_i, δ_i are the individual thesaurus, document collection, set of queries, and mapping from queries to document sets of each local IR system. The whole distributed IR system can be redefined as $S = (T, D, Q, \delta)$ where $T = \bigcup_{i=1}^{n} T_i, D = \bigcup_{i=1}^{n} D_i$, and

$$Q \supset \bigcup_{i=1}^{n} Q_i, \preceq_j = \preceq \bigcap (Q_j \times Q_j)$$

which means the queries can be obtained by combining the queries at each local site. Moreover, the partial ordering at each site j only pertains to the queries at site j. As for each query in the grand system, the document collection for a query contains the documents whose descriptors are at least as specific as the query.

$$(\forall q \in Q)(\forall d \in \delta(q) \in 2^D, q \preceq t(d))$$

Based on this model, the hierarchy represented by γ is established and partitioned among the different sites. A local site at a lower hierarchy is called a *subsystem* of a higher one if it satisfies several specific criteria. [13] A query sent to the distributed IR system is then forwarded to the local *subsystem*s where a local query is performed. The local responses are afterwards sent back to the originating site where the final result set is combined from the local ones. For example, if S_1 is a *subsystem* of S_0, then the query results at site S_0 contain those found in S_1:

$$\forall q \in Q, \delta_1(q) = \delta_0(q) \bigcap D_1$$

2.3 Problems of Existing Web Search Systems

Web IR systems, usually referred as Web search engines, are special IR systems, which can be built in a centralized or decentralized fashion. They are quite different from traditional IR systems mainly in the size of the document set, the organization of the set (ad hoc but linked by Web links vs. mostly independent), and the way the document set is built (by crawlers vs. according to specific criteria chosen by the people preparing and collecting the documents). For Web search engines, ranking computation of documents is a key component to return results highly satisfying users' information needs as searchers are usually only interested in the top few retrieved documents. There are decentralized search systems studied and built according the distributed IR model, including meta-engines and research-oriented prototypes, which however never reached the level comparable with non-meta engines. In general, large-scale experiments have not been seen using these approaches so their effectiveness and efficiency for Web search engines remain unknown. The classical model of distributed search systems briefed above is suitable for traditional information systems such as library, static collection of medical documents, etc.. When dealing with information seeking in Web search engines for the following reasons, it has some original sin because proper partition of a hierarchy is extremely important in this model, otherwise the resulting hierarchical subsystems may not be valid in the sense of returning correct search results, namely the result-containing property may be broken. [13] The reason behind is that the partitioning of the Web is not controllable by people. We can not re-organize the whole Web according to *Document partitioning* or *term partitioning* as we wish and do in a traditional distributed IR model. We believe these are the reasons why main Web search engines take a centralized architecture and mainly rely on global ranking algorithms. Global ranking algorithms, e.g, Google's PageRank, for centralized search engines, have been extremely succussful as people have known. However, global ranking of documents has two potential problems: high computation cost and

potentially poor rankings. Both of the problems are related to the centralized computation paradigm. [5] There are also more specific problems because of the unique properties of the Web:

1. Coverage studies show that a small percentage of Web pages are in all search engines. Moreover, fewer than 1% of the Web pages indexed by AltaVista, HotBot, Excite, and Inforseek are in all of those search engines. [8] This fact also justifies the use of meta search engines, which however never reached the level of success comparable with non-meta engines.
2. It is likely that the larger the indexed subset of the web, the higher the recall and the lower the precision, for a given query. Query-based search engines still return too much hay together with the needle. One possible reason accounting for this is the current ranking algorithm is not really capable of differentiating the Web documents in the huge Web collection pertaining to the queries.
3. On the other hand, Web directories do not have enough depth to find the needle. The reason is that they are usually compiled manually or semi-automatically thus the timeliness and availability are largely limited. A reasonable decentralized architecture will enhance the situation greatly.

Thus we propose to decentralize the task of searching and ranking. In our work, first of all we introduce a ranking algebra providing such a formal framework. [5] Through partitioning and combining rankings, we manage to compute document rankings of large-scale web data sets in a localized fashion. Secondly we propose innovative ways of computing Web document rankings based on ideas inspired by Swarm Intelligence. [6] Thirdly we put dynamic interactions among the Web servers in our architecture that enables the decentralized Web search system to compute timely and accurate global rankings in a Peer-2-Peer fashion. We achieve initial results, demonstrating that the use of such an approach can ameliorate the above-mentioned problems. The approach presents a step towards a decentralized search architecture for Web and P2P systems.

3. Existing Works and Their Limitations

Research on distributed IR systems has not been limited to the abstract model. Running systems were also built to realized the previously proposed ideas. In these systems both engineering issues common to distributed systems and algorithmic issues specific to IR need to be taken care of.

3.1 Harvest

Harvest [7] is a distributed crawler-indexer architecture which addresses the main problems in crawling and indexing the Web: Web servers get requests from different crawlers of search engines which increase the servers' load; most of the entire objects retrieved by the crawlers are useless and discarded; no coordination exists among the crawlers. But it seems most of further Harvest applications are in the field of caching Web objects instead of providing advanced internet search services. State of the art indexing techniques can reduce the size of an inverted file to about 30% of the size of the text (less if stopwords are used). For 100 million pages, this implies about 150GB of disk space. Assuming that 500 bytes are required to store the URL and the description of each Web page, we need 50GB to store the description for 100 million pages. The use of meta search engines is justified by coverage studies that show that a small percentage of Web pages are in all search engines. Moreover, fewer than 1% of the Web pages indexed by AltaVista, HotBot, Excite, and Inforseek are in all of those search engines. [8]

3.2 WAIS

Wide Area Information Service (WAIS) [9] is a very early piece of work in the area of web-based distributed query processing. It was popular at the beginning of the 1990s before the boom of the Web. A WAIS system only forwards queries to certain servers based on a preliminary search of the content of those specific servers. The servers use some special fields in the documents such as *headline* of a news article or *subject* of an email to describe the content. This approach serves as a compromised solution between forwarding the request to all servers, and forwarding the request to only those servers that match the very detailed full-text index.

3.3 GLOSS

The work of Glossary-of-Servers Server (GLOSS) [10] builds a server to estimate the best server for a given query based on the vector-space model. Each individual server is characterized by its particular vector. The top n servers are then searched and the results are combined. The work explored several means of characterizing a server. It is estimated that the index on the GLOSS server is deemed to be only 2 percent of the size of a full-text index.

3.4 STARTS

Stanford Proposal for Internet Meta-Searching (STARTS) [11] is a protocol for distributed, heterogeneous search. It was designed from scratch to support distributed information retrieval and includes features intended to solve the algorithmic issues related to distributed IR, such as merging results from heterogeneous sources.

3.5 Z39.50

Z39.50 [12] is a standard for client/server information retrieval which defines a widely used protocol with enough functionality to support most search applications. It was firstly approved as a standard in 1995 but is under revision recently. The protocol is intended to query bibliographical information using a standard interface between the client and the host database manager which is independent of the client user interface and of the query database language at the host. The database is assumed to be a text collection with some fixed fields. The protocol is used broadly and is even part of WAIS. Not only the query language and its semantics, but also the way of establishing a session, communication, and exchange of information between client and server are specified in the protocol. It was originally conceived only to operate on bibliographical information, but has been extended to query other types of information as well.

3.6 Modern P2P Systems

Modern P2P systems developed very quickly in recent several years. Search functionalities in these systems however are really preliminary and limited. Most use the naive way of broadcasting requests such that the whole P2P network is flushed. And no systematic and mature public search engine like the Web counterpart Google appears yet. This leaves much space for us to study and integrate the requirements into our architecture.

4. My Approach: The Architecture

In our architecture, we introduce the logical abstract *aggregator* of the processing units of a decentralized search system. A picture of aggregator graph is illustrated here.

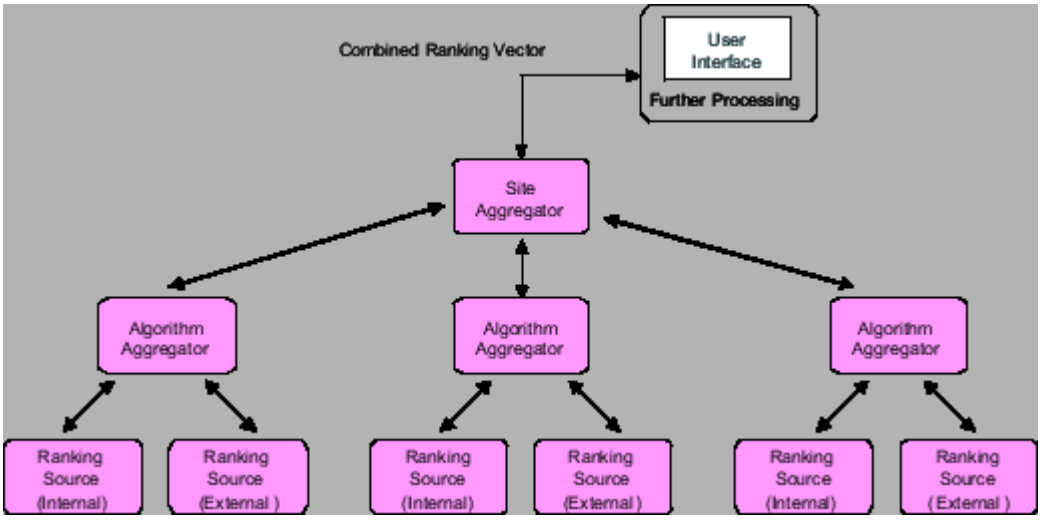

Figure 1: Aggregator Graph

We have 3 types of different roles in our architecture. *User* is the first role who sits on top of all and submits queries to member(s) of a decentralized search system. Original ranking *sourcer* is the role locating at the bottom that provides original ranking vectors of source documents. In the middle is the role *aggregator* which takes input from original ranking sources or intermediate aggregators and compute an aggregated ranking vector according to a ranking algebra. This newly computed ranking vector can be in turn exported to higher level aggregators as their input. There are two types of aggregators: a *site* aggregator or an *algorithm* aggregator. A site aggregator combines the results of different sites while an algorithm aggregator combines the results of different algorithms. Details about the ranking algebra and interactions among sites in the architecture are elaborated in the following subsections.

4.1 Rank Composition: The Algebra

In our experiments we found [5] that different rankings established in different contexts (in particular local vs. global contexts) can be of great interest. Thus we deviate from the view of usual Web search systems that all documents are ranked within a single, absolute ranking. Rather we see rankings as first-class objects, that can be produced, exchanged and manipulated as any other data object. To make this precise, we introduce now a framework for ranking computation that defines what the type of rankings is, and how rankings are manipulated. We will use an algebraic framework for rankings, a ranking algebra, similarly as it is done for other types of data objects (such as using relational algebra for relations). The ranking algebra will allow to formally specify different methods of combining rankings, in particular, for aggregating global rankings from local rankings originating from different semantic contexts. We define the domain of objects that are to be ranked. Since rankings can occur at different levels of granularity there will not be rankings of documents only, but more generally, rankings over subsets of documents (partitioned zones). In order to be able to compare and relate rankings at different levels of granularity we introduce now a partial order

on partitions. We also introduce an operator to make it possible to directly relate the elements of two partitions to each other (and not only the whole partitions as with cover). Link matrices defined over partitions are the basis for computing rankings. A number of operations are required to manipulate link matrices before they are used for ranking computations. We introduce only those mappings that we have identified as being relevant for our purposes. The list of operations can be clearly extended by other graph manipulation operators. We also need the ability to change the granularity at which a link matrix is specified. This is supported by the contraction operator. In certain cases it is necessary to directly manipulate the link graph in order to change the ranking context. This is supported by a link projection. Normally rankings will be normalized. As for link matrices we also need to be able to project rankings to selected subsets of the Web. In many cases different rankings will be combined in an ad-hoc manner driven by application requirements. We introduce weighted addition for that purpose. After having these necessary definitions, we can apply the ranking algebra to produce different types of rankings by using different ranking contexts such as:

- Global site ranking: The global site ranking is used to rank the selected Web sites using the complete Web graph.
- Local site ranking: In contrast to the global site ranking we use here as context only the subgraph of the Web graph that concerns the selected Web sites.
- Global ranking of documents of a Web site: This ranking is the projection of the global PageRank to the documents from a selected site.
- Local internal ranking for documents: This corresponds to a ranking of the documents by the document owners, taking into account their local link structure only. The algorithm used is PageRank applied to the local link graph.
- Local external ranking for documents: This corresponds to a ranking of the documents by others. Here for each document we count the number of incoming links from one of the other Web sites. The local links are ignored.

Now that we have seen different ways to derive rankings using the ranking algebra, we illustrate of how these rankings can be combined in order to produce further aggregate rankings. This will be again specified by using ranking algebra expressions. Thus we address several issues that have been discussed in previous sections and demonstrate two points:

1. We show that global document rankings can be determined in a distributed fashion, and thus better scalability can be achieved. Hence ranking documents based on global information not necessarily implies a centralized architecture.
2. We show how local rankings from different sources can be integrated, such that rankings can be made precise and can take advantage of globally unavailable information (e.g. the hidden web) or different ranking contents. Thus a richer set of possible rankings can be made available.

The application of the ranking algebra to compute the rankings occurs at both the site aggregators and the algorithm aggregators.

4.2 Local Ranking in the Dynamic Web Society

Traditional ranking models used in Web search engines rely on a static snapshot of the Web graph, basically the link structure of the Web documents. However, visitors' browsing activities indicate the importance of a document. In the traditional static models, the information on document importance conveyed by interactive browsing is neglected. The nowadays Web server/surfer model lacks the ability to take advantage of user interaction for

document ranking. We enhance the ordinary Web server/surfer model with a mechanism inspired by swarm intelligence to make it possible for the Web servers to interact with Web surfers and thus obtain a proper local ranking of Web documents. The proof-of-concept implementation of our idea demonstrates the potential of our model. The mechanism can be used directly in deployed Web servers which enable on-the-fly creation of rankings for Web documents local to a Web site. The local rankings can also be used as input for the generation of global Web rankings in a decentralized way. This innovative way of computing local Web document rankings is used at the level of *sourcer* in the aggregator graph. We use Web *pheromone* to record users' visiting information which reflect how interesting or how important a Web document is from the viewpoints of the surfers. Whenever a surfer accesses a page, some Web pheromone is left on the page. The Web server assumes the role of the Nature and maintains the Web pheromone information of all the local documents. Pheromone accumulation, evaporation, and spreading strategies are defined and applied to all documents. The higher pheromone density a document has, the more important in a general sense it has and thus the higher it would be ranked. In our model, the Web surfers here are the natural agents in a self-organizing system just like the ants in their social intelligent system. [14] Surely surfers are not non-intelligent, but as human have only really limited insight on the Web and most of the time can only follow the hyper links created by someone else without any knowledge of the structure of the Web graph, so here we the surfers as the primitive agents that abide by simple operation rules. Furthermore, the Web server here is not only a passive listener to the requests for Web documents, but also an active participant of the self-organizing system by assuming the role of an arbiter who assures the rules are carried on during the interactive interactions between the requesting visitors and the requested Web documents which form together the ecological environment for the self-organizing system.

4.3 Gluing All Together: Global Site Ranking

Here we go a further step to introduce the ranking of Web sites in the global Web. We have already the way that every local Web site can use to generate absolutely timely local Web document ranking with potentially high quality. We also have a ranking algebra that an aggregator can use to combine inputting rankings to get the intermediate and final ranking result. But we still lack one thing: how the local rankings can be compared with each other? How are the computed float values from different local Web sites interpreted when putting together? Global site ranking is our criteria as the answer. Just like Web documents, Web sites are also considered to have different degrees of general importance. Nobody will deny the higher importance of Yahoo! or Google or W3C Web sites. By computing global site ranking we get the base where the construction of aggregators can be built on. What is required for the computation of global site ranking is the knowledge of link structure among the sites on the Web. We will study on two sub problems:

1. How the knowledge is exchanged and shared by all participating Web sites?
2. How big a part of the Web graph of which the link structure information is needed for a Web site to compute an approximation of global site ranking that is good enough?

As a beginning, we will let the Web sites use the naive broadcasting way to exchange and share the knowledge of link structure among sites. As the number of Web sites is much much smaller than the number of Web documents, the computation of ranking is definitely tractable at the scale of sites. In the future, we may study more efficient ways of synchronizing the knowledge of the global Web among the Web sites. The second sub problem is more complex. We are thinking about identifying a subset of more critical sites in the computation of the approximate global site ranking for every particular Web site. Then this critical subset

is used to hopefully have a good enough approximation of the real global site ranking based on the whole Web graph.

5. Results Achieved So Far

5.1 Ranking Algebra

We apply the ranking algebra in a concrete problem setting. We performed an evaluation of the aggregation approach described above within the EPFL domain which contains about 600 independent Web sites identified by their hostnames or IP addresses. We crawled about 270.000 documents found in this domain. Using this document collection we performed the evaluations using the following approach: we chose two selected Web sites with substantially different characteristics, in particular of substantially different sizes. For those domains we computed the local internal and external rankings. Then we applied the algebraic aggregation of the rankings obtained in that way, in order to generate a global ranking for the joint domains. For local aggregation we chose a higher weight (0.8) for external links than internal links. One motivation for this choice is the relatively low number of links across subdomains as compared to the number of links within the same subdomain. The resulting aggregate ranking for the joint domains is then compared to the ranking obtained by extracting from the global ranking computed for the complete EPFL domain (all 270.000 documents) for the joint domains. The comparison is performed both qualitatively and quantitatively. We have better qualitative results. In the top 25 list of the aggregate ranking result, the top 4 are obviously more important than the top listed results from the global PageRank. We can assume that this is an effect due to the agglomorate structure of these document collections. These play obviously a much less important role in the composite ranking due to the way of how the ranking is composed from local rankings. It shows that the global page ranking is not necessarily the best possible ranking method. We obtained similar qualitative improvements in the ranking results of other domains. As for quantitative results, one can observe that basically the aggregate ranking approximates the rankings computed on the selected subsets. This is an interesting result, since the aggregate ranking is performed in a distributed manner, computing separate rankings for each of the subdomains involved. This shows that by aggregation one can obtain at least as good results in a distributed manner as with global ranking using the same information. Details are included in the paper [5].

5.2 Swarm Intelligent Web Server Module

We developed a swarm intelligent module for the popular Apache Web server software. Although it still has bugs which make the server instable and crash from time to time, we did manage to make some preliminary but very interesting experiments with it. Firstly we made a game of Quest for Treasure. The idea was to start a quest for a treasure in order to see, if in a self-organised system, changes to the environment will result in a collective optimisation of navigation. We had 12 rooms where the visitors could navigate through. In two of these rooms were treasure chests, which had to be explored. Above each button was the actual pheromone density (on the right side of the vertical bar) of the underlying link. Visitors had to use for the navigation the density which was computed and shown by the server module. Red numbers remind people that the pheromone there has very high density. After a certain time we could just follow the red links and we found the treasure. The next morning we removed one of the treasure chests, and we could observe that during the next few hours the colors had changed. On the way to the room where we removed the chest the density of pheromone was decreased and the red links again led now to the one and only chest. So we could observe a very simple form of self-organization by collectively using the Web pheromone information from the

server module. Hence, we demonstrate that the swarm of internet surfers is indeed more intelligent than a single surfer. Then we did a small-scale experiment with a lab Web site which has about 200 Web pages in order to explore this possibility of generating document ranking from our swam intelligent module, which we call *intelligent* ranking. We installed the module and requested volunteers to surf the site. The experiment lasted for 2 days (because of the instability of the module). We pre-computed the static PageRank ranking. We found that the PageRank ranking and the intelligent ranking is quite different. Firstly, the top ranked document is different; Secondly, in the top 17 documents of both rankings, only 6 (35%) are the same. More details can be found in [6].

5.3 Simulation of Cooperative Web Servers

We are investigating proper simulation environments for Web and P2P systems. Factors taken into consideration include scalability, which is probably the most important for a Web scale problem; ease of development, for example, the language used, the modularity, the interface definitions, etc.; administrative capabilities in order to monitor the simulation, customize parameters and settings, observe the progress of execution, gather statistical information and results; visualization; etc.. After that, we will develop the prototype system of our decentralized architecture for Web and P2P search engines and implement the algorithms that we have briefly discussed. We will focus on the cooperative interactions among the Web servers.

6. Conclusion

6.1 Summary

As the global site ranking is more or less stable because at a grand scale the Web is more or less stable although there are fluctuations because of the continuous of emerging and dying Web sites. Thus the global site ranking only needs to be computed periodically like what is done by modern search engines for computing the document rankings of the whole Web. The ranking algebra makes it possible to represent formally the decomposing the computation of the global document ranking to two step: the computation of local document rankings for every Web site; and then the combination of the computed local rankings. Swarm intelligence has been reported to have applications in many fields, such as combinatorial optimization, communication networks, robotics, etc.. As far as we know, nobody else or other research groups have tried to apply this idea for Web surfing to obtain ranking of documents for the purpose of information retrieval. By developing a swarm intelligent module, we turn a Web server into a self-organizing component of the aggregator graph in the Web. Combining the evaluation information implied in surfers' dynamic interactions with the Web server, we might have a ranking of local Web documents of better quality from the viewpoints of users. In short, by adopting my decentralized architecture, the computation cost of Web and P2P search engines will be reduced dramatically; the timeliness of search results of Web documents is enhanced a lot without suffering the crawling delay of nowadays Web search engines; potentially results fit more for the searchers' information needs will be returned thanks to the innovative way of computing local document rankings.

6.2 My Contributions

My work is original as the Swarm Intelligence-inspired way proposed by me of computing the document ranking of a Web site is an innovative new idea; and I also propose the completely new method of computing the global document ranking of the whole Web from the global site

ranking and local document rankings in a totally decentralized way. Our work of ranking algebra is also original since it provides a formal framework which is absent in most ad-hoc systems for computing document rankings.

References

1. Karl Aberer. "P-Grid: A Self-Organizing Access Structure for P2P Information Systems". 2001.
2. Sergey Brin, Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". 2000.
3. Larry Page, Sergey Brin, R. Motwani, T. Winograd. "The PageRank Citation Ranking: Bringing Order to the Web". 1998.
4. Jon Kleinberg. "Authoritative Sources in a Hyperlinked Environment". 1998.
5. Karl Aberer, Jie Wu. "A Framework for Decentralized Ranking in Web Information Retrieval". APWeb2003.
6. Jie Wu, Karl Aberer. "Swarm Intelligent Surfing in the Web", submitted to ICWE2003.
7. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. "The Harvest Information Discovery and Access System". In *Proc. 2nd Int. WWW Conf.*, pages 763-771, Oct. 1994.
8. Ricardo Baeza-Yates, Berthier Ribeiro-Neto. "Modern Information Retrieval". Addison-Wesley, 1999.
9. Duda, A. and Sheldon, M. (1994). "Content routing in a network of WAIS servers". In *Proceedings of the IEEE Fourteenth International Conference on Distributed Computing Systems*, pages 124-132.
10. Gravano, L. and Garcia-Molina, H. (1995). "Generalizing GLOSS to vector-space databases and broker hierarchies". In *Proceedings of the 21st International conference on VLDB Conference*, pages 78-89.
11. Luis Gravano, Chen-Chuan K. Chang, and Hector García-Molina. "STARTS: Stanford proposal for Internet meta-searching". In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 207-218, Tucson, AZ, May 1997.
12. NISO press, Bethesda, MD. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-200X)*, 2002. Accessible at http://www.niso.org/.
13. David A. Grossman, Ophir Frieder. "Information retrieval: algorithms and heuristics". Kluwer Academic Publishers, Boston, MA, 1998.
14. Eric Bonabeau, Marco Dorigo, Guy Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, 1999.

Footnotes

... engines.[1]

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

... keywords[2]

The concept of *descriptor* is used for these keywords in some reference. Accordingly an *asciptor* is defined as a term that is a synonym of a descriptor. We do not cover the details here.