Distributed Community Crawling

Fabrizio Costa
Department of Computer Science
Università degli Studi di Firenze, Italy
costa@dsi.unifi.it

Paolo Frasconi
Department of Computer Science
Università degli Studi di Firenze,Italy
paolo@dsi.unifi.it

ABSTRACT

The massive distribution of the crawling task can lead to inefficient exploration of the same portion of the Web. We propose a technique to guide crawlers exploration based on the notion of Web communities. The stability properties of the method can be used as an implicit coordination mechanism to increase the efficiency of the crawling task.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - clustering, search process, information filtering; H.5.4 [**Information interfaces and presentation**]:
Hypertext/Hypermedia - navigation

General Terms

Algorithms, Experimentation

Keywords

Distributed Crawling, Web Metrics, Web Communities

1. INTRODUCTION

The rapid growth of the Web makes efficient document indexing and retrieval increasingly difficult if implemented in a centralized fashion. A possible solution lies in the distribution of the crawling task among a high number of simultaneous and coordinated processes. The coordination mechanism can increase the overall efficiency if it can avoid that different crawlers acquire the same documents. With no coordination each crawler would explore its own queue of URLs, ignoring that other crawlers might have already traversed that of portion the Web [2]. This event would not be infrequent, especially when the exploration is guided by link score algorithms (such as PageRank or HITS) as pages having high score will be more likely to be retrieved by many crawlers. The coordination mechanism is therefore crucial to the efficiency issue, but if not enough attention is paid to its costs it can easily constitute a penalizing overhead as the crawlers could end up spending much time and bandwidth exchanging URLs or enquiring on the presence of URLs in each other list. In order to reduce the overhead we suggest to use an implicit coordination mechanism based on the concept of Web communities, that is sub-graphs of the Web that are characterized by documents that preferentially link documents belonging to the same sub-graph [5, 3]. In this work we introduce the Community Crawling algorithm whose stability properties are a first step towards building a massively distributed and efficient crawling architecture.

2. COMMUNITY CRAWLING

The concept of Web communities has been formalized in many ways [5, 3]. Here we use a similarity measure to establish the degree of membership of a new item with respect to a collection of documents immersed in a web of citations. This measure will drive the community crawling algorithm. We represent a set of documents as nodes and their citations as edges in a directed graph $G = (V, E)$. In addition to the notion of direct link between two nodes $p, q \in V$, $l(p, q) = 1/2$ if (p, q) or $(q, p) \in E$ and $l(p, q) = 1$ if $(p, q) \wedge (q, p) \in E$, we make use of two kinds of indirect relations: co-citation (when there exists a node k that links both p and q) and bibliographic coupling (when there exist a node k that is linked by both p and q). We speak of degrees of co-citation and bibliographic coupling to express the strength of these indirect relations. More precisely, let C_p and C_q be the set of documents linked by p and q respectively. The bibliographic coupling degree $f(p, q)$ is defined as $f(p, q) = |C_p \cap C_q|$. Equivalently we introduce the co-citation degree $c(p, q)$ as $c(p, q) = |P_p \cap P_q|$ where P_p and P_q are the set of documents that link p and q respectively. The overall similarity measure between p and q is defined as $s(p, q) = \alpha \cdot l(p, q) + \beta \cdot f(p, q) + \gamma \cdot c(p, q)$ where the coefficients weight the relative importance of each contribute to the total similarity measure (in the reported experiments they were all set to one). We now introduce the measure of similarity, that we call *membership*, between a document and a set of documents. The membership of p with respect to a set R is obtained collecting the k most similar elements of R in $S_k(p, R)$ and taking the average similarity of $S_k(p, R)$, that is $m_k(p) = \frac{1}{k} \sum_{q \in S_k(p,R)} s(p, q)$. When $k = |R|$ we obtain the overall average similarity of p. In this case the membership measure might be dominated by many not very similar documents which can lead to numerical precision problems in ranking operations. Setting $k = 1$ makes the measure very sensitive to local elements and might drive the clustering algorithm to build clusters that are not very "compact". Optimal values for k can be determined experimentally. The membership measure allows us to identify documents that are highly connected to the set that represents the current estimate of the community in $\Theta(V^2 log V)$. We now extend the notion of membership in order to better account for the initial lack of information about the real community. It seems reasonable to assume that the importance of documents which are topologically near nodes with high membership should be increased even if there are no other documents in the estimated community that exhibit a similar pattern of citations. We therefore spread the membership value of each node to its neighbors in a discounted fashion as $n_k(q, p) = m_k(p)i^d$ where $m_k(p)$ is the mem-

WWW2004, May 17–22, 2004, New York, NY USA.
ACM 1-58113-912-8/04/0005.

bership value of node p, i is a discount factor and d is the length of the shortest path between q and p. Each node finally acquires a total membership score of $M_k(q) = m_k(q) \sum_{p \neq q} n_k(q,p)$. The algorithm starts from each node of G and performs a breadth-first visit of the neighborhood, leading to an additional cost of $\Theta(V^2 + VE)$. To decrease the algorithmic complexity we adopt two strategies: we consider the influence only of a fraction of the nodes (those with a higher membership value) and we spread the membership only to nodes that are connected with paths of a maximum predefined length. Choosing a fixed number of representatives per community, and assuming that the number of outgoing links does not depend on the size of the community, we obtain a complexity $\Theta(V)$.

The Community Crawling algorithm uses the membership similarity measure to perform graph clustering. The algorithm takes as input the number of communities, their maximum estimated size and an initial collection of documents. Each community is managed by a single process (crawler) with its own exploration queue and is independent of the other processes. The exploration queue contains the cited documents that have not yet been acquired ranked by the membership value of the citing document. When the capacity of the crawler is reached, documents with a low membership value are discarded. This procedure increases the overall intra-cluster similarity and keeps the exploration focused on a specific community.

3. EXPERIMENTAL RESULTS

We are interested in evaluating the clustering and stability properties of the Community Crawling algorithm, that is, how much each crawler is capable to focus on a single community and retrieve most of its documents, avoiding to explore regions of the graph that contain documents that are not relevant for that community. We run a set of experiments taking Breadth-First Crawlers as a baseline competitor algorithm since this simple strategy (as shown in [4]) retrieves important nodes during the very early stages of the exploration and has a null coordination costs.

One of the main problems in assessing the clustering performance of exploration algorithms on the real Web is that we do not have a trustworthy strategy to determine which are the true communities even if we can use the entire graph. We therefore resort to artificial graphs whose structure mimics that of the Web and that consists of intentionally built clustered regions. In our experiments the Web is therefore made of sub-graphs, each representing a community, loosely connected with each other. The entire set of nodes V is partitioned into sub-sets $V_t = v_i^t$ where $i \in 1, .., n_t$. Chosen a parameter k, edges are generated as $E_t = \{(v_i^t, v_j^t) | j = (i + z + n_t) \mod n_t\}$ for each $v_i^t \in V_t$ and $z \in [-k/2, k/2]$. In this way nodes share cited and citing neighborhoods. Finally, edges connecting different communities are introduced: each node links o nodes in other communities. The variation of the ratio k/o allows us to model communities with a varying degree of separation; from perfectly separated ($k/o \to \infty$) to completely intermixed ($k/o = 0$).

We generated artificial Web graphs of 1700 nodes, each graph contained 17 communities of 100 nodes each. Every node had $k = 32$ and $o = 2$. In the experiments a system consting of 16 Community Crawlers was tested against 16 Breadth-First Crawlers. The capacity of each crawler was set to 200 nodes. In order to test the stability properties of the two different crawling strategies, we measure the average cohesion, the precision and recall of the retrieved sub-graphs. The cohesion of a sub-graph is the ratio between the number of edges that connect nodes within the sub-graph and the total number of edges of the sub-graph. The precision (recall) is the ratio between the number of true nodes belonging to a community and the number of nodes in the sub-graph (the number of nodes in the real community). We run two kind of experiments in order to observe the behavior of the algorithm under different initial conditions. When initializing each crawler with a set of elements extracted from the entire graph in a random fashion, we obtain for the Community Crawler sub-graphs with an average cohesion of 76.2%, a precision of 37.7% and a recall of 75.4%, while for the Breadth-First Crawlers we have cohesion 56.7%, precision 33.1% and recall 66.3%. In a second experiment we initialized each crawler with elements coming only from a specific community and we obtained for the Community Crawler an average cohesion of 83.9%, a precision of 42.9% and a recall of 85.8%, while for the Breadth-First Crawlers cohesion was 50%, precision was 36% and recall was 70%. The results indicate that the Community Crawling strategy retrieves highly interconnected sub-graphs that are representative of each community. We have also noted that even when the initialization is performed in a random manner each crawler retrieves documents only from a few communities.

A set of preliminary experiments have also been run on a sample of the Web of on one million documents (WebTrack10g [1]). In this case we do not have information of real communities and hence we cannot measure precision/recall values. In the experiment 16 crawlers were used initializing them with 10 Web sites chosen at random that had the word "genetic" in their URL and their capacity was set to 10,000 nodes. Membership was spread at a maximum neighborhood distance of 2 (discount factor 1/2), and $k = 10$. The cohesion of the retrieved regions was 89.8% for Community Crawlers against 18.6% for Breadth-First Crawlers, which supports our intuition that the concepts introduced in this work remain valid when applied to the real Web graph.

4. CONCLUSIONS

We have introduced a parallel exploration/clustering algorithm that focuses on highly clustered parts of a graph. The use of this technique can be a first step towards building massively distributed Web crawling systems that avoid retrieving the same documents at low inter-process communication costs.

5. ACKNOWLEDGMENTS

This work was supported in part from the Italian Department of Education and Research (MIUR) under Grant 2003091149 and under the Grid.it Project.

6. REFERENCES

[1] Commonwealth Scientific and Industrial Research Organization. In *http://es.csiro.au/TRECWeb/*.

[2] J. Cho and H. Garcia-Molina. Parallel crawlers. In *Proc. of the 11th International World–Wide Web Conference*, 2002.

[3] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[4] M. Najork and J. L. Wiener. Breadth-First Crawling Yields High-Quality Pages. In *Proceedings of the 10th International World Wide Web Conference*, pages 114–118, Hong Kong, May 2001. Elsevier Science.

[5] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the web. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*. ACM Press, 1996.