A P2P-based Incremental Web Ranking Algorithm

Sumalee Sangamuang  Pruet Boonma  Juggapong Natwichai
Computer Engineering Department
Faculty of Engineering, Chiang Mai University, Thailand
sangamuang.s@gmail.com, {pruet, juggapong}@eng.cmu.ac.th

Abstract—In this paper, we propose an incremental algorithm for web ranking in the Peer-to-Peer (P2P) environments. Not the same as the non-incremental algorithm, the proposed algorithm can partition the web link graphs, the graphs represented the connectivity structure among the web pages, into the changed subgraphs, and the unchanged subgraphs. Subsequently, the algorithm processes only the necessary data in order to compute the ranking. The experiments have been conducted to evaluate the efficiency of the algorithm, comparing with the non-incremental algorithm in various P2P environments. We report the efficiency in terms of both computational and communication costs. It has been found that in terms of communication cost, the proposed algorithm can out perform the traditional one in all configurations. For the computational cost, the proposed algorithm can outperform the traditional one. In order to investigate this issue, in this paper, the web-link graph size is increased. The very large web-link graph allows the authors to clearly understand the benefit/shortcoming of the algorithm.

The organization of this paper is as follows. The proposed incremental algorithm on P2P models both in terms of the intuition and the design details is presented in the next section. In Section III, the experiment results to evaluate our work are reported. The related work is presented in Section IV. Finally, Section V gives the conclusion and outline of the future work.

I. INTRODUCTION

Web ranking is one of the most important components for web search engines [1]. After the web crawlers store the web page content into the repository, the engines analyze such content to compute the ranking such that the most relevant web pages are ranked in the descendant order. Among the ranking model, Google’s PageRank [1] is one of the most notable models. In such model, both the forward and back links are considered based on the randomly page-traversal. The forward links distribute the “importance” to the target pages in part. While the backward links obtain importance from the other pages. The link connectivity of the collected web pages forms the web-link graph in which the PageRank algorithm is to be applied iteratively until the PageRank values are converged.

Different computing platforms are proposed for addressing the web ranking problem, since its iterative nature. Peer-to-peer (P2P) computing is one of the most promising approaches which can improve the efficiency of the web ranking processes [2], [3]. Since such problem can be addressed in a divide-and-conquer basis, in which each node can compute its corresponding sub web-link graphs.

However, as the collected web content continues to grow when a new web-site is identified by the crawlers. The needs to provide the fresh search results to the users are being increased by their changing behavior considering personal and business necessity. Thus, re-computing the web ranking every time the new pages are collected can be inefficient, even in the efficient computing platforms.

In this paper, we propose an incremental algorithm to compute the web ranking based on a P2P computing platform. Instead of re-computes the web ranking, our proposed algorithm can process only the changed subgraphs. It start with partitioning the web link graphs into the changed subgraphs, and the unchanged subgraphs. Subsequently, the algorithm processes only the necessary data in order to compute the ranking. With this way, though the complexity is not decreased, the non-worst case execution time can be more efficient in the real-world situation.

The PageRank model, as well as the JXP algorithm [2] which is presented. In Section A, we begin with presenting the PageRank model, as well as the JXP algorithm [2] which is our foundation. Subsequently, our approach is presented in Section B.

A. Preliminaries

A PageRank [?] value of a web page, p, is defined as in Equation 1.

\[
\text{PageRank}(p) = \epsilon \times \sum \frac{\text{PageRank}(q)}{\text{Outdegree}(q)} + (1 - \epsilon) \times \frac{1}{N} \tag{1}
\]

q is a webpage which links to p. The Outdegree function is the number of outgoing link of the webpage. N is the number of web pages in the considering web-link graph. The \( \epsilon \) is the random traversal probability, usually, it is set at 0.85.

Typically, the PageRank values of each web page in the web-link graph is computed iteratively until the average PageRank values is converged to a pre-specified value. As mentioned before, the PageRank computing itself might not have a high-computational cost. However, the need to refresh the ranking result as fast as possible encourages that an efficient way to compute the PageRank is required.
In [2], JXP algorithm is proposed to compute the PageRank values on the P2P computing models. Such algorithm will segment the whole web-link graphs into multiple subgraphs. Each subgraph will be assigned to a peer in the P2P network. In some situations where the multiple web crawlers work independently, each crawler may feed its explored web-link graph as the subgraph directly.

For each peer, the PageRank of each web page is computed locally in the same way as the traditional PageRank computing. In each local web-link graph, a special node, so called “world-node” is added to represent the connectivity information of the outside peer. Let Internal be a set of the pages within a local web-link graph. Since the value of the PageRank is converged. The weight of the world-node \( w \) of each local web-link graph can be calculated as shown in Equation 2.

\[
\text{PageRank}(w) = (1 - ( \sum_{i \in \text{Internal}} \text{PageRank}(i)))
\]  

(2)

Subsequently, some nodes in a local web-link graph which have an incoming link from the world-node will obtain weight from the world-node as calculated using Equation 4. Intuitively, a page will obtain a fraction of the PageRank from the world-node.

\[
\text{weight}(w \rightarrow w_i) = \frac{1}{\text{PageRank}(w)} \times \sum_{w \rightarrow w_i} \frac{\text{PageRank}(w_i)}{\text{Outdegree}(w)}
\]  

(3)

Nodes in a local web-link graph which have an incoming link from the world-node will obtain PageRank from the world-node as calculated using Equation 4. Intuitively, a page will obtain a fraction of the PageRank from the world-node.

\[
\text{weight}(w \rightarrow w_i) = \frac{1}{\text{PageRank}(w)} \times \sum_{w \rightarrow w_i} \frac{\text{PageRank}(w_i)}{\text{Outdegree}(w)}
\]  

(4)

When the local graphs at each peer have been processed, a core peer (manually selected) will issue the peer meeting process to another peer. Subsequently, such two peers will exchange their PageRank values. Typically, a peer with the high different PageRank is preferable since the two peers can benefit from the meeting process more. The PageRank computing is finished after all the peers have exchanged their information.

B. Improvement

With regard to the original JXP algorithm, when some web pages are discovered by the crawlers, the whole local PageRank will have to be re-computed. Also, the peer meeting will have to re-process afterward.

Thus, we propose our algorithm for such incremental issue, incremental page-ranking algorithm (ICR). The proposed algorithm composes of two components as in the JXP algorithm. The first part computes the local PageRank of each peer. Subsequently, the second algorithm merges the local web-link graph from all the peers together. The details of each algorithm are as follows.

The incremental local-PageRank computation is presented in Figure 1. In the algorithm, the PageRank of each node in the web-link graph is initialized, if there is no value from the previous iteration. Subsequently, the web-link graph is segmented into two partitions, i.e the changed and the unchanged partitions with respect to the weight and the structure of the graph. The segmentation can be computed by comparing the nodes of the existing graph with the new one. After the web-link graph is segmented, the set of boundary nodes between the two partitions is determined. The PageRank of such set will only be affected from the changed partition, not the structure. After adding the world-node into the set, the PageRank of the set is re-computed at the end.

The merging algorithm is presented in Figure 2. The algorithm begins with the core peer to be merged with the others. When two web-link graphs are to be merged, the nodes and the links of such graphs are combined straightforward. If there is a duplicate node in the two graphs, the average weight of them will be assigned. Once the merged graph has been determined, it will be segmented and computed as in the local PageRank algorithm.

The efficiency of the proposed incremental algorithm is depended on the increment data. When the P2P networks for such processing is large the increment data will be relatively small, and thus the convergence of the PageRank will be able to reach more rapidly. Also, in such cases, the communication cost will be relatively low. Such claims will be validated by the experimental results in the next section.

III. Experimental Results

This section discusses a simulation experiment evaluating the proposed approach. The proposed approach, the ICR algorithm, is compared with the JXP algorithm [2]. The evaluation focuses on two performance metrics, the number of computation rounds for local graph merging and the number of transmitted messages between peers. The first metric shows the performance of the proposed approach in terms of computational cost, compared with the JXP, while the second metric shows the communication cost. The metrics are evaluated with the varying sizes of P2P network and web link graph.

A. Experimental Configuration

The simulation was performed on PeerSim, a P2P simulator using Chord P2P network[7]. The largest network size is 1,800 peers. The data set used in the experiments comes from Amazon.com as in [2]. This data set is a web-link graph which contains up to 100,000 nodes. In each experiment, the average result from multiple runs are presented.
A P2P-BASED INCREMENTAL WEB RANKING ALGORITHM

1 Let \( p \) be a peer in the P2P network.
2 Let \( L_{\text{wlg}} \) be a local web-link graph of size \( n \) nodes.
3 Let \( G_{\text{wlg}} \) be the global web-link graph of size \( N \) nodes.
4 Let \( w \) be the world-node in the peer.
5 If \( \text{PageRank}(w) = \text{Null} \) then
6 Set \( \text{PageRank}(w) = (N - n) / N \), \( n_i \) is the \( i \)-node in \( L_{\text{wlg}} \).
7 End If
8 For \( i = 1 \) to \( n \)
9 If \( \text{PageRank}(n_i) = \text{Null} \) then
10 Set \( \text{PageRank}(n_i) = 1 / N \).
11 End If
12 End For
13 Find the changed partition of \( L_{\text{wlg}} \), denoted as \( cL_{\text{wlg}} \).
14 Find the unchanged partition of \( L_{\text{wlg}} \), denoted as \( uL_{\text{wlg}} \).
15 Find the set of boundary nodes between \( cL_{\text{wlg}} \) and \( uL_{\text{wlg}} \), denoted as \( \text{adj}L_{\text{wlg}} \).
16 Add \( w \) into \( \text{adj}L_{\text{wlg}} \).
17 Determine the PageRank in \( \text{adj}L_{\text{wlg}} \).

Fig. 1. Incremental Local PageRank Algorithm

1 Let \( \text{largest}_{\text{wlg}} \) be the largest local web-link graph.
2 Let \( G_{\text{weight}} \) be the collection of the weight of all webs.
3 Let \( m \) be the number of peers in the P2P network.
4 For \( i = 1 \) to \( m \)
5 Select a local web-link graph \( b_{\text{wlg}} \) from a peer randomly.
6 Let \( m_{\text{wlg}} = \text{Merge}(\text{largest}_{\text{wlg}}, b_{\text{wlg}}) \).
7 Find the changed partition of \( m_{\text{wlg}} \), denoted as \( cm_{\text{wlg}} \).
8 Find the unchanged partition of \( m_{\text{wlg}} \), denoted as \( um_{\text{wlg}} \).
9 Find the set of boundary nodes between \( cm_{\text{wlg}} \) and \( um_{\text{wlg}} \), denoted as \( \text{adj}m_{\text{wlg}} \).
10 Add \( w \) into \( \text{adj}m_{\text{wlg}} \).
11 Determine the PageRank in \( \text{adj}m_{\text{wlg}} \).
12 Update the processed PageRank in \( G_{\text{weight}} \).
13 Disconnect \( b_{\text{wlg}} \) from \( m_{\text{wlg}} \).
14 End For

Fig. 2. Incremental Merging Algorithm

B. Experimental Results and Discussion

Figure 4 shows the experimental result which represents the impact of web-link graph size on the communication cost. The number of peers in the P2P network is five peers. From the original web-link graph with 46,505 nodes, a fraction of the web-link graph is obtained to represent a smaller web-link graph. Thus, on the x-axis, the web-link graph size varies from 9,301 to 46,505 nodes. On the y-axis, the number of transmitted messages is shown. The number of transmitted messages, representing communication cost, is measured when the peers perform peer meeting to merge local graph.

From the figure, the ICR has lower communication cost, up to 50%, in all web-link graph size compared with the JXP. Also, the differences is large when the web-link graph size increases. Because the ICR considers only the nodes that are affected by adding nodes, thus, smaller size of data is required to be exchanged among peers.

Fig. 3. Impact of web-link graph size on the communication cost

Fig. 4. Impact of web-link graph size on the computational cost

Figure 3 shows the experimental result with a setup similar to Figure 4; however, on the y-axis, the number of round to perform graph meeting instead of the number of transmitted messages. Also, the number of web-link graph
is increased to up to 100,000 nodes.

Figure 4 shows a experimental result which represents the impact of web-link graph size on the computational cost. The P2P network size is 5 peers. From the original web-link graph with 100,000 nodes, a fraction of the web-link graph is obtained to represent a smaller web-link graph. Thus, on the x-axis, the web-link graph size varies from 12,000 to 100,000 nodes. On the y-axis, the computational cost is shown in the number of rounds to complete graph merging.

From the figure, ICR has lower, up to 15%, computational cost compared with JXP when the web-link graph size is less than 60,000 nodes. However, from the size of 60,000 nodes upward, ICR has slightly higher, up to 5%, computation cost. This indicates that when the web-link graph size increases, ICR uses more round to perform graph merging. This is because when the web-link graph size is larger, each peer also has larger number of outgoing edges to the world nodes. Even though ICR uses comparatively fewer nodes for calculation, but it has to handle many outgoing edges to the world nodes. Thus, the same or larger number of rounds is required. This small additional cost can be seen as a shortcoming of the proposed approach, however, when consider the huge improvement of communication cost, the authors believe the additional cost is acceptable.

Figure 5 shows the experimental result which represents the impact of the P2P network size on the computational cost. The P2P network size, on the x-axis, varies from 10 to 50 peers. The web-link graph size is 12173 nodes. On the y-axis, the computational cost is shown in the number of rounds to complete graph merging.

From the figure, ICR has lower computational cost, up to about 50%, in all P2P network sizes compared with JXP. Again, because ICR only considers a fraction of nodes affected by adding nodes, it requires smaller number of messages to be exchanged. The result validates the claim in Section II-B.

The experiment results indicate that ICR has smaller computational and communication costs compared with JXP in the most configurations. Only when the web-link graph is very large, ICR has slightly larger computation cost. However, the huge improvement of communication cost makes the larger computation cost an acceptable trade-off. In terms of web ranking results, the difference between the two approaches is very small. However, because of the limited space, the results are not included in this paper.

IV. Related Work

The typical framework for search engines compose by the web crawlers, indexers, and ranking components. After the web crawlers gather the content of the web pages to the repository, the indexers will analyze the important features of the webs, e.g. URL, header, meta-tag, bold letters, or links. Link structures between the pages form a web-link graph which is an input for the web ranking computing. One of the most notable approaches to rank the web is Google’s PageRank [1]. The concept of the PageRank is based on the randomly traversal on a graph considering the back-link of each web-page as its PageRank value. While, the forward-links distribute the PageR-
ank value of the other pages. Another important approach is the Hubs and Authority model [5]. In the Hubs and Authority model, the back-links and the forward-link of a page are also considered. However, the forward-link is considered as the hub value, while the back-links as the authority value. A web page can be categorized into hub or authority based on such two values.

For the incremental graph processing, there exist several proposed work. For example, an approach for the graph incremental correlated-change detection problem is proposed in [6] as follows. First, the given graph is segmented to determine the change part. Subsequently, the similarity between subgraphs is computed. Last, the correlated-changes can be detected using the similarity. In [7], an approach to incremental compute the PageRank is proposed. The approach begins with partitioning the given web-link graph into two segments, i.e. the unchanged segment, and the increment segment. Subsequently, only the PageRank of the pages associated with the increment segment can be computed. Our web-link graph segmentation is based on such work.

For the applying P2P computing models to improve the efficiency of the web ranking, it was first addressed in [2]. In such work, each computational peer contains a local web-link graph, and also computes the web ranking locally, a web-page is represented as a node, and a link is represented as an edge. In [3], a heuristic approach to improve the peer-meeting processes is proposed. Instead of selecting a peer to be met randomly, a peer will select another peer which has high “authority” quality, i.e. high number of outgoing links. To evaluate the authority of a peer, the general statistic information such as average or standard-deviation can be applied. Such information will be used for the evaluation subjected to minimize the network load. In [8], the authors proposed an orthogonal, but closely related to the web ranking computation, i.e. to rank the web community content using PageRank authority style in P2P networks. The authority content can be considered as a trend in such context. Such work can be applied in the social network environment.

V. Conclusion

In this paper, we have proposed an efficient incremental algorithm, the ICR algorithm, to compute web ranking. Such algorithm composes of the two main algorithms, local web ranking in each peer, and web ranking information exchanging between the peers. In particular, at each peer, when a new set of webs emerges, the local web ranking algorithm will consider the incremental changes in term of the linkage structure. The experimental results on the real data set have shown that the proposed work can significantly reduce both the computational cost as well as the communication cost in the most cases. Only when the web-link graph is very huge, the proposed work has slightly higher computation cost compared with the traditional algorithm. However, the huge improvement in communication cost is an acceptable trade-off to the computation cost increment. In the future work, we will focus on develop-

VI. ACKNOWLEDGMENT

We would like to thank Faculty of Engineering, Chiang Mai University, Thailand, as well as the Graduate School for their financial support.

REFERENCES