웹 크롤러를 위한 효율적인 URL 우선순위 할당 기법*

Md. Hijbul Alam, 하종우, 조윤호, 이상근
고려대학교 정보통신대학 컴퓨터통신공학부
e-mail : {hijbul, okcomputer, cloudjo21, yalphy}@korea.ac.kr

Efficient URL Prioritizing Method for Web Crawlers

Md. Hijbul Alam, Jong-Woo Ha, Yoon-Ho Cho, SangKeun Lee
Division of Computer and Communication Engineering, Korea University

요 약

With the amazing growth of web faster important page crawlers poses great challenge. In this research we proposed fractional PageRank, a variation of PageRank computed during crawl that can able to prioritize the downloading order. Experimental results shows that it outperforms the prior crawler in terms of running time yet provide a well download ordering.

1. Introduction

The web is growing in astonishing rate because of various newly available framework of publishing content. However the users only see the topmost 10-30 pages for their corresponding query which demands that crawler should give high priority to important pages for downloading and save resource by not downloading the less important pages. State of the art technique that deal with important page crawling need significant amount of running time. Hence crawling important pages early draw new attention that internal computation of prioritizing the discovered pages should be fast enough to keep up the phase with downloading pages that needs external resources such as network bandwidth and so on.

Most of the Prior works of crawling important page mostly deal with how well it selects important pages early during downloads [2][3][8][10]. In addition to that Dr. Cho et. al. proposed RankMass Crawler and Windowed RankMass Crawler in [9] that use PageRank lower bound to prioritize the URL and showed tradeoff between the quality of crawling important page early and the performance overhead involve in term of running time. In our simulation we found RankMass crawler is not able to download the web pages speedy compare to the quality they compromise. Conversely our crawlers capable to good give ordering for prioritize the WebPages and many times fast than prior one.

In crawling process there are different phases such as discovering URL, downloading and exploring page. We compute fractional PageRank of a page in every state particularly and use this value to prioritize downloading of WebPages. In short the contribution of this research is as follows:

- We proposed new algorithms to prioritize downloading WebPages that achieves high performance gain.

2. Related Work

URL prioritization for crawling is a well studied problem. Most of the previous studies don't focused on the running time rather than they only try to maximize PageRank coverage [2][3][8][10]. However with the amazing growth of the web size it's now become an important issue. For example IRLbot crawl 6.3 billion pages within 41 days and the number of discovered link is 41 billion [11]. Hence prioritize the discovered link efficiently is very important factor. RankMass Crawler was the first attempt to provide a tight guarantee on the PageRank coverage of the downloaded pages during download [9]. However this approach introduces huge performance overhead. As mentioned by [9] to simulate RankMass (RM) crawler for downloading 80 million pages (with 141 million links) pages it needs 222 hours. Therefore some approximation of original algorithm has been proposed by authors which is known as Windowed-RankMass algorithm that reduce the overhead by batching together sets of probability calculations and downloading sets of pages at a time which also consume significant amounts of local resource.

3. Measuring Importance of Web

PageRank is very effective and most useful global measure of page quality since its invention to ranking the Web Pages. We use alternative method of computing PageRank that has been proven in [7] and also presented in [9]. In the following we give short description of this algorithm because in our proposed method we need path probability which is a component of PageRank.

$$PageRank(p_i) = \sum_{p_j \in D} \sum_{w_{ji} \in W_{ji}} Path\,\mathrm{Pr}obability(w_{ji}) \quad (1)$$

PageRank of page p_i can be expressed as summation of the

* 이 연구에 참여한 연구자는 '2 단계 BK21'의 지원비를 받았음

probability of each path w_{ji} in set W_{ij} that lead from page p_j to page p_i, for every page p_j in the whole web D is given in equation 1, where, the probability of one specific path w_{ji}, PathProbability(w_{ji}) can be formulate as equation 2 through random surfer model [6]

In random surfer model a random surfer interrupted with probability *1-d* randomly jumped to a page, p_j with probability t_j continued clicking only one link per page (having O_k outlinks) with probability $1/O_k$ without interruption (with probability d^c, where c is number of pages in sequence w_{ji}, i.e., the number of links the random surfer clicks in the session).

$$Path\,\Pr obability(w_{ji}) = \prod_{p_k \in D} \frac{1}{O_k}(1-d)t_i d^c \quad (2)$$

4. Proposed Method

We compute Fractional PageRank (FPR) that is summation of path probability in the following algorithm using equation 2 for a certain period during crawling and use this to prioritize the web pages.

Definition 1: Fractional PageRank (FPR) is defined as the summation of path probability value a page receives after discovering and before downloading the pages.

Algorithm: We have design a new algorithm that use fractional PageRank to prioritize the queue. That is the discovered link with highest fractional PageRank will be downloaded first. The algorithm is presented in figure 1.

```
1.  fractionalPageRankCrawl()
2.      for each page    in the set of trust seed page do
3.              fpr_i = ( 1 – d ) * t_i
4.
5.      while(Queue is not empty)
6.          pick p_i with largest fpr_i
7.          download p_i if not downloaded yet
8.          for each p_j linked to by p_i and p_j is not downloaded do
9.              fpr_j = fpr_j +(d* fpr_i)/o_i ;
10.         fpr_i =0
```

(**Figure 1**) Fractional PageRank algorithm

As we can see fractional PageRank algorithm in figure 1 fractional PageRank Crawler model the behavior of random surfer with personalized vector. Hence it is strong enough to combat with spam. Though this algorithm is not iterative but combining the behavior of random surfer makes it powerful enough to prioritize the web. The size of discovered link that will be crawl next located in the frontier grows up very quickly in large crawl. As the number of edges in the web graph is very large using RankMass crawl is impractical and in RankMass or Windowed RankMass crawl whenever a new inlink to a page is discovered the page is explored again and again to compute the exact PageRank lower bound of a page. However fractional PageRank crawl a new inlink is contributed fractional PageRank if the webpage is not downloaded yet i.e as soon as if a page is downloaded, the incoming links to this page will be discarded. Though Fractional PageRank crawler don't consider some prestige by discarding some incoming links, we found that still it's able to download the web page with well download order which is described details in next section.

5. Evaluation

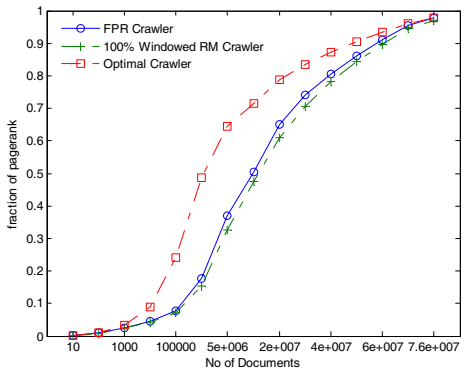
We simulate our algorithms on sub graph of web of 80,644,902 pages and 2,481,281,617 links that correspond to pages under the .uk top domain namely uk-2006-06. The graph is freely distributed in [1][4]. The pages are obtained by UbiCrawler [5] in June 2006 with maximum 16 depth per host and 50000 pages per host. For our simulation we have used only a single machine Intel Quad 2.4 Ghz processor with 4 GB RAM in Solaris environment. We have done all the processing in the main memory.

We use two metric to evaluate our algorithm. First, how much Cumulative PageRank is collected during crawl. Second, the running time required for important page crawling.

Algorithm	Time	No of pages downloaded
FPR Crawl	83 minutes	76,688,586
100 % Windowed RM	5:20 hours	75,000,000

<**Table 1**> Algorithm performance by running time

As mentioned in related work that RankMass algorithm takes so long time Windowed RankMass algorithm was proposed. 100% Windowed RankMass algorithm is the fastest important crawler among all the Windowed RM which takes 5:20 hours to download 75 million pages. However from table 1 we can see that FPR Crawler outperforms by taking only 83 minutes and choosing well download order.



(**Figure 2**) Cumulative PageRank Collected by different FPR Windowed RM and Optimal Crawler

From our experiment we found that FPR Crawler behave almost same as 33% Windowed RM. A possible explanation of this phenomenon is Fractional PageRank consists of a large portion of PageRank lower bound which able to chose high PageRank pages first with high probability. The optimal curve shows the Cumulative PageRank collected by an ideal crawler. An ideal crawler tries to maximize the Cumulative PageRank collected in each download and is assumed to know all the URLs and the PageRank value of the pages in the frontier.

6. Conclusions

In this paper we examine how to speed up prioritizing the URL maintaining well chosen download order. We see the algorithm not only outperforms by running time but also download the web in up to standard order.

References

[1] Laboratory for Web Algorithm. http://law.dsi.unimi.it/

[2] Serge Abiteboul, Mihai Preda, and Gregory Cobena: Adaptive on-line page importance computation. In *Proceedings of 12th international conference on World Wide Web*, pages 280–290, 2003.

[3] Ricardo Baeza-Yates, Carlos Castillo, Mauricio Marin, Andrea Rodriguez, Crawling a country: Better strategies than breadth-first for web page ordering, *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 864-872, 2005.

[4]Paolo Boldi, S. Vigna, The webgraph framework I: Compression techniques, In *Proceedings of the 13th international conference on World Wide Web*, pages 595-602, 2004.

[5] Paolo Boldi, Bruno Codenotti , Massimo Santini , Sebastiano Vigna, UbiCrawler:A scalable fully distributed web crawler, *SoftwarePractice & Experience*, 34(8):711-726, 2004.

[6] Sergey Brin, Lawrence Page, The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, 30(1-7):107-117,1998.

[7] Michael Brinkmeier, PageRank revisited, *ACM Transactions on Internet Technology (TOIT)*, 6(3):282-301, 2006.

[8] Junghoo Cho , Hector Garcia-Molina , Lawrence Page, Efficient crawling through URL ordering, In *Proceedings of the 7th international conference on World Wide Web*, pages 161-172, 1998.

[9] Junghoo Cho, Uri Schonfeld, RankMass crawler: A crawler with high personalized pagerank coverage guarantee, In *Proceedings of the 33rd international conference on Very large data bases*, pages 375-386, 2007.

[10] Marc Najork , Janet L. Wiener, Breadth-first crawling yields high-quality pages, In *Proceedings of the 10th international conference on World Wide Web*, pages 114-118, 2001.

[11] Hsin-Tsang Lee and Derek Leonard and Xiaoming Wang and Dmitri Loguinov. IRLbot: Scaling to 6 billion pages and beyond. In *Proceeding of the 17th international Conference on World Wide Web*, pages 427-436, 2008.