

Distributed Ranking over Peer-to-Peer Networks

Tak-Pang Lau

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
tplau1@cse.cuhk.edu.hk

Dexter Siu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
cwsiu1@cse.cuhk.edu.hk

ABSTRACT

Query flooding is a problem existing in Peer-to-Peer networks like Gnutella. Firework Query Model solves this problem by Peer Clustering and routes the query message more intelligently. However, it still contains drawbacks like query flooding inside clusters. The condition can be improved if the query message can send directly to the query destination, as the message does not need to send hop by hop. This can be achieved by ranking. By ranking, the network can know the destination and the information quality shared by each peer. We introduce distributed ranking in this paper. We give background of FQM, outline of the proposed method, and conduct a series of experiments that demonstrate the significant reduction of query flooding in a P2P network.

Categories & Subject Descriptors: H.3.3

[Information Storage and Retrieval]: Information Search and Retrieval – Search Process; H.3.4 [Information Storage and Retrieval]: System and Software – Distributed Systems;

General Terms: Algorithms, Design

Keywords: Distributed Peer Ranking, Peer-to-Peer Networks

1. INTRODUCTION

Firework Query Model (FQM) was proposed by our group to reduce the query flooding problem in a typical P2P network such as Gnutella [1]. It is based on peer clustering by using attractive links to form localized clusters of similar information content. One of the problems with FQM is that within the cluster there is still query flooding. There are several ways to solve this. One is to randomly select neighbors to send query message. The Second is to increase the number of attractive links. However, these are not good. The first method can miss some important peers contain valuable content. The second method requires severe bandwidth requirement. In this paper, we propose to use distributed ranking to solve the localized cluster query flooding problem. This method can reduce the query messages produced and also retain its quality.

2. DISTRIBUTED RANKING

Our algorithm is based on the Distributed PageRank Calculation (DPRC) [3], which is used in WWW page ranking. We use DPRC as our basis because of two reasons:

- It is already used in P2P environment
- It is already a distributed algorithm

In DPRC, all the web pages in WWW are partitioned into several groups. A page ranker is assigned to each group to handle the page ranking calculation of pages within the group. In order to share information between different page groups, page rankers are linked up like a P2P networks. Figure 1 shows a page group with four pages.

From Figure 1, the links between pages are classified into 4 types. The first type is *Inner Link*, which connects pages within the group. The second type is *Virtual Link*, which is inserted to avoid link sink problem in PageRank. The third type is *Afferent Link*, which is the link pointed from other pages group to inside. The fourth type is called *Efferent Link*, which is the link pointed from pages inside the group to outside.

Base on the link types, PageRank of a page can be divided into three parts:

- PageRank coming from pages within the groups
- PageRank coming from virtual links
- PageRank coming from pages in other group

Page ranker in each group first refresh the rank coming from other groups. Then it calculates the rank similar to the original PageRank algorithm in [2]. After that, it calculates the outgoing rank and sends it to other groups' page rankers.

The remaining questions are how to partition peers into different groups, and how to assign peer ranker. It is because DPRC is already designed for used in P2P networks formed by page rankers. The following are the main considerations:

- **Fairness**--In P2P networks, each peer should be treated in the same way. It is unfair for some nodes to perform ranking, but not for others.
- **Overlapping**--As ranking scores are globally scaled, if the ranking scope of two peers overlaps, resources are wasted.
- **Peer Communication**--Under DPRC, peer communications are very important for transferring the ranking to other groups. If ranking needs to be transferred to a peer ranker which is far away, the generated packet will cause the scalability problem.

Base on the above consideration, our proposed algorithm makes the "peer group" only consist of one peer, and that peer is the ranker itself. In other words, each ranker only needs to rank itself. By partitioning in this way, fairness can be achieved, no overlapping problem exists and peer communication only limits to the attractively connected neighbors. The modified algorithm is as follow:

Algorithm Distributed Peer Ranking

```
DPRC()
R0 = S %%initial ranking
X = 0 %%incoming rank from other group
loop
  Xi+1 = Refresh X
  Ri+1 = (1-c) + Xi+1
```

```

for each attractively connected Peers u
  calculate (Rank / out-degree) and send to u
end
Wait for some time
while true

```

Each peer continues to run the algorithm. Finally, their own score becomes static. The score can then be broadcasted to the whole network in the form of Gnutella's ping-pong message. As the score is globally scaled, the comparison between two peers can be directly determined by the score.

After performing distributed ranking, we can direct query message to the target destination using the ranking result. This saves the message needed for traveling hop by hop.

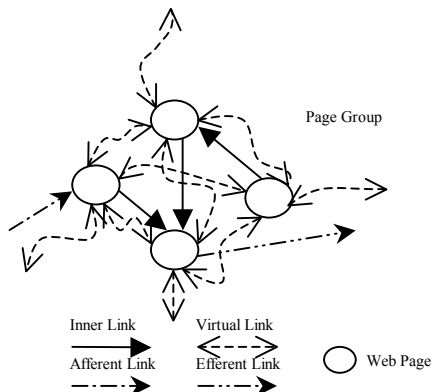


Figure 1. A web page group

3. EXPERIMENTAL RESULT

We verify our proposed method by comparing the convergence speed of and the query reply speed in our proposed distributed ranking with the centralized approach. The centralized approach is that each peer acts as a "peer crawler". It ranks the neighbor by getting their link status. Then it runs the "PageRank" algorithm to get the ranking result. For convergent speed, it is measured by the number of iterations for the rank value to become static. With respect to the number of generated packets, we measure it in one iteration of the methods. For query reply speed, it is measured by counting the number of hops needed for the reply message of query back to the query source. We test it under Sun Blade 100 machine using language C and Matlab.

From Figure 2, we observe that distributed ranking needs much fewer iterations in order to make the estimation come to static value. It is because for centralized ranking, each time the peer just crawls one peer, while in distributed ranking, every peer cooperates, and distributes the score to its neighbors at the same time. As a result, distributed ranking needs much fewer iterations.

From Figure 3, we find that distributed ranking requires smaller minimum reply path length than FQM. It is because the ranking mechanism can directly reply with the query message with peer list instead of passing messages within cluster. The query has shorter reply path length have faster retrieval speed. The messages travel through less peer when the query have shorter minimum path length.

4. CONCLUSION

In this paper, we introduce the distributed ranking method. This technique allows peers to share the computational and storage costs. We verify our proposed method by performing simulations and show that our method converges faster than the centralized ranking.

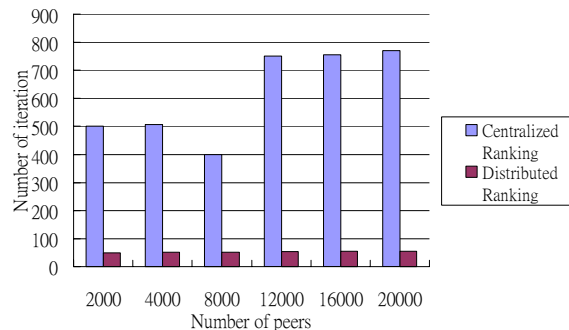


Figure 2. Convergence Speed (Number of Iteration) Against Network Size

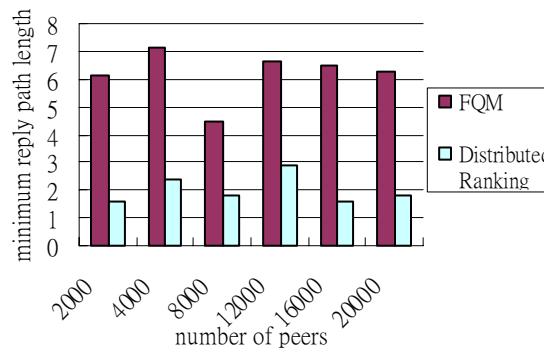


Figure 3. Minimum Reply Path Length against Number of Peers

5. ACKNOWLEDGEMENTS

This work is supported in part by the Earmarked Grant, CUHK #4351/02E, from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region

6. REFERENCE

- [1] Irwin King, Cheuk Hang Ng, and Ka Cheung Sia. Distributed content-based visual information retrieval system on peer-to-peer networks. ACM Transactions on Information Systems, accepted.
- [2] S. Abiteboul, G. Cobena, M. Preda. Adaptive On-Line Page Importance Computation. 12th International World Wide Web Conference, 2003, Budapest, Hungary, 20-24 May, 2003.
- [3] ShuMing Shi, Jin Yu, GuangWen Yang, DingXing Wang. Distributed Page Ranking in Structured P2P Networks. Proceedings of the 2003 International Conference on Parallel Processing (ICPP '03)