iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VII **Month of publication:** July 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Reducing Web Response Time in Peer to Peer Network Using Peer's History and Caching

Anusree K S[1], S Sindhu[2]
Dept of Computer Science & Engineering,
NSS College of Engineering, Palakkad, Kerala

Abstract— With the dramatic explosion of online information, the Internet has undergone a transition from a data communication infrastructure to a global information utility and web searching has become a normal way of information access. But normal web search query process takes a lot of time to respond to user queries in the event of high network traffic or flash crowd queries. Traditional method to deal with such a situation is to cache the web pages in the local systems and share it with others who are requesting for the same web page. But caching all the web pages accessed by a system is a complex process and is not always necessary. This paper proposes a novel method that uses a combination of a peer's browsing history and caching to optimize web search. A peer's browsing history is exploited to retrieve the recently accessed web pages corresponding to repeated identical queries from same of different user. Further, the event of flash crowd in the network is detected using a change detection algorithm and the corresponding web page is cached in the local system's cache to be shared with other systems requesting for the same web page.
Keywords— Web search optimization, Web page caching, Peer to Peer network, Flash crowd

I. INTRODUCTION

Peer to Peer (P2P) network is a popular technology used for sharing and searching files on the computers connected to a network. With the emergence of P2P file sharing applications, millions of users have used P2P network to search desired data. A P2P network has also shown a great potential to become a popular network tool for sharing information on the web, where information resides on millions of sites in a distributed manner. The phenomenal growth of the Internet as an information source makes web content distribution and retrieval one of its most important applications today. The increasing amount of information in the web, as well as the number of new users who are inexperienced in the art of web search, creates new research challenges in the field of information retrieval such a relevance of information retrieved and its access speed [2]. Normal web search query process takes a lot of time to respond to user queries in the event of high network traffic and flash crowd queries. Web caching is a widely deployed technique to reduce the latency observed by web browsers [7]. But caching web pages all the time is unnecessary and complex. Also the growth rate of web content is much higher than the rate with which memory sizes for web caches are likely to grow [8].

This paper proposes a novel web search optimization technique for peer to peer networks that uses the browsing history of the peers to retrieve the answers to repeated identical queries under normal traffic conditions and uses web page caching in the event of flash crowd queries. In this way the proposed work intends to reduce the time to access relevant information and the complexity of caching the web pages all the time.

II. RELATED WORKS

A. Flash Crowd

A flash crowd is a sudden surge in traffic to a particular web site that causes the site to be virtually unreachable. Flash crowds happen when many end users simultaneously send requests to one web site usually because of special events attracting interest of the mass population. These events could be pre-scheduled such as a webcast of a popular movie or they could be unpredictable like natural disasters, breaking news and links from popular web sites (that is, the "slash-dot effect").When a flash crowd event occurs, the volume of requests to the target web server increases dramatically. The magnitude may be tens or hundreds of times more than normal conditions. These requests may overload network connections and the target server. In the meanwhile, response traffic could also congest networks (often in the first few links where traffic concentration is the largest). As a result, most users perceive unacceptably poor performance [2]. Major news web sites experience this problem during major world events. Flash crowds often

cause very poor performance at the server side and result in a significant number of unsatisfied clients. In addition, flash crowds unintentionally deny services to other end-users who either share common network links with flash crowd traffic or retrieve unrelated information from the target server. Handling flash crowds poses a difficult task for web services. Content Distribution Networks (CDNs), hierarchical web caches, and P2P networks have all been proposed as mechanisms for mitigating the effects of these sudden spikes in traffic to under provisioned origin sites [3]. Traditional way to deal with flash crowd queries is to cache the data at local sites [1]

B. Web Caching

Web page caching aims to reduce network traffic, server load, and user-perceived retrieval delay by replicating popular content on proxy caches that are strategically placed within the network at organizational boundaries, for example . Caching of web contents (e.g., HTML pages, images) can reduce bandwidth usage, server load, and user's from proxy caches, which are located more closely to users than web server[7]. Web contents can have an expiry time associated with them after which the object is considered to be stale. A stale object is not used. If the object in the cache is stale, then it is equivalent to the object not being in the cache. An expiry date can be specified in the http header of a web object [5]. Network infrastructure vendors such as Akamai deploy web caches and CDN to protect servers from overloading during flash crowds. These services need infrastructure support and are usually expensive. There are two major types of Web caches, a browser cache and a proxy cache. A browser cache is a part of all popular Web browsers. The browser keeps a local copy of all recently displayed pages, and when the user returns to one of these pages, the local copy is reused. By contrast, a proxy cache is a shared network device that can undertake Web transactions on behalf of a user, and, like the browser cache, the proxy cache stores the content. Subsequent requests for this content, by this or any other user, will trigger the cache to deliver the locally stored copy of the content, avoiding a repeat of the download from the original content source [9].

There are many caching strategies used to optimize web search in peer to peer network [1, 3, 7, 8, 9] . Caching in search engines has been studied on two levels. The first level of caching, namely the result caching, takes place at the front end and deals with the case where identical queries are issued repeatedly by the same or different user. Thus, by caching a few ten thousand to a few million results that have recently been returned by the engine, we can filter repeated queries from the workload and increase overall throughput. At the second level, list caching is used. List cache keeps inverted lists corresponding to frequently used search terms in the main memory, resulting in additional benefits for engines with disk-based index structures [7].

One such method has been proposed by Erika Rosas, Nicolas Hidalgo, Mauricio Marin, Veronica Gil-Costa in [1]. It uses a two level caching which uses a result cache and location cache at each peer. Each peer maintains a local result cache used to keep the answers for queries originated in the peer itself and queries for which the peer is responsible for by contacting the Web search engine on-demand. To achieve a short-term fair distribution of queries there is a location cache in each peer which keeps pointers to peers that have already requested the same queries in the recent past. This lets these peers share their query answers with newly requesting peers.

In paper [2], a three-level caching architecture with an additional intermediate level of caching has been proposed. This intermediate level, called intersection caching or projection caching (depending on the implementation), caches inverted list data for pairs of terms that commonly occur together in queries with more than two search terms. The basic idea relies on the fact that all of the major search engines by default only return documents that contain all of the search terms. Thus, search engines need to score only those documents that occur in the intersection of the inverted lists. Unfortunately, in most cases the most efficient way to find the intersection still involves a complete scan over the lists, and this dominates the cost of query processing. By caching pairwise intersections between lists, which are typically much smaller than each of the two lists, the cost in subsequent queries can be significantly reduced.

Network Early Warning System (NEWS)[3], proposes a router based adaptive admission control mechanism to detect the occurrence of flash crowd event in the network. It employs a technique that detects flash crowd from performance degradation in web response and mitigate it by admitting incoming requests adaptively. It has two novel aspects. First, NEWS does not consider per-flow service requirements for incoming requests like most admission control schemes. Instead, it determines end-user perceived performance through measurement. Second, NEWS monitors changes in the performance of high-bandwidth responses because of their sensitivity to overloading conditions. Based on this observation, NEWS adjusts the admitted request rate automatically and adaptively. These two approaches make NEWS a self-tuning system, adaptive in both server- and network-limited scenarios A fundamental problem that confronts peer-to-peer applications is to efficiently locate the node that stores a particular data item.

Volume 3 Issue VII, July 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Chord [4], a distributed lookup protocol that addresses this problem, provides support for just one operation: given a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key/data item pair at the node to which the key maps. Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is continuously changing. Chord uses a variant of consistent hashing to assign keys to Chord nodes. Consistent hashing tends to balance load, since each node receives roughly the same number of keys, and involves relatively little movement of keys when nodes join and leave the system. Each Chord node needs routing information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other node.

III. PROPOSED SYSTEM

The existing systems for web search optimization in peer to peer network caches all the web search results of the users for efficient retrieval of the data, especially in the events of flash crowds [1,6]. But the fact is that a flash crowd event doesn't occur very frequently. So caching all the results always become complex and unnecessary. The proposed system uses the searching history of the other peers in the P2P network to load the web search results for repeated identical queries. Whenever a user submits a web query to the browser, the query is stemmed to get the keywords and matched against already searched queries in the network. If a match is found, i.e., another user in the network had already retrieved result to a similar query, then URL of web page accessed by the other user for the corresponding query is loaded on to the current user's system directly. If a query is given for the first time then the user is directed towards a normal Google search [Figure 1]. This system is intended to reduce not only the memory usage but also reduces the time to get the results compared to normal web search. This method will also help users who are inexperienced in the art of web search to retrieve results that seemed relevant to other users for the same or similar query. In order to cope with flash crowd queries, the proposed system will detect the events of flash crowds using a change detection algorithm. The change detection algorithm will detect a flash crowd by observing the change in web response time. When a flash crowd event is detected, the proposed system will cache specific result in the local systems so that it can be shared with other peers requesting for the same page thus reducing load at the web server.

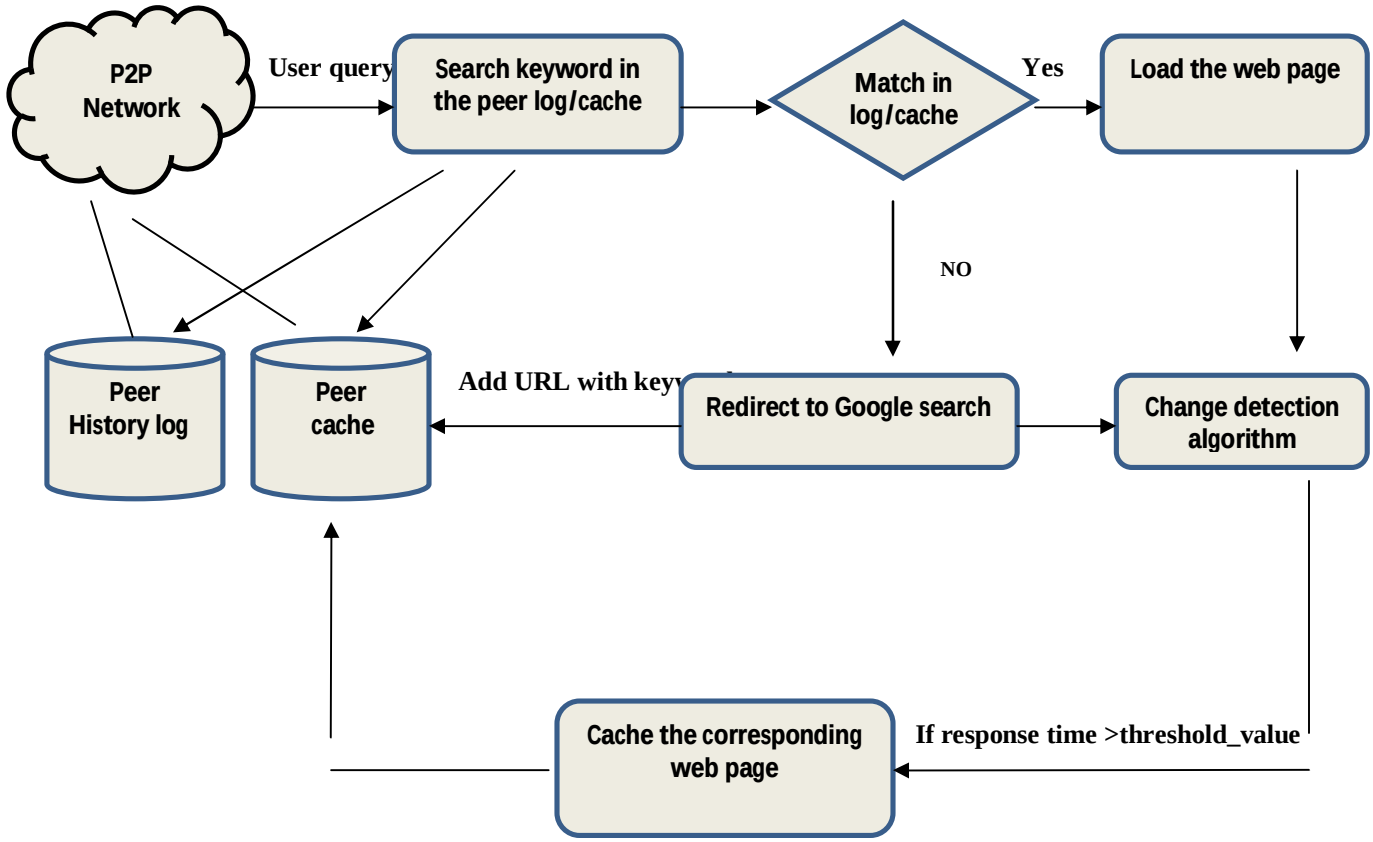


Fig1: Flow diagram of the proposed system

IV. CONCLUSIONS

The study proposes a novel method for reducing user perceived delay in accessing information in peer to peer networks which uses a combination of web page caching and peer browsing history. The scheme uses browsing history of peers in the P2P network to answer the similar queries of a new user so that the user is able to retrieve a web page seemed relevant to another user corresponding to the query. Since the web query of new user is answered with the help of browsing history of other peers who already requested the same query, there is no additional overhead of caching or indexing of all web query results as in the existing systems. The proposed system not only reduces the overhead and response time but also reduces the traffic towards the search engine. The contributions of the proposed system are mainly as:

A strategy that quickly reacts to user queries, even in the event of a sudden and drastic increase in query traffic.

A strategy that requires web pages to be cached only in the event of flash crowds reducing the portion of main memory required for storing cached pages.

Using these two strategies in combination the proposed system can efficiently cope with web query traffic in a peer to peer network

V. ACKNOWLEDGEMENT

The authors would like to thank professors of NSSCE, Palakkad for suggestions and support on this paper.

REFERENCES

[1] Erika Rosas, Nicolas Hidalgo, Mauricio Marin, Veronica Gil-Costa, “Web search results caching service for structured P2P networks, Future Generation Computer Systems,” Jan 2014, pp. 221-231.

[2] Xuan Chen and John Heideman, “Flash Crowd Mitigation via Adaptive Admission Control Based on Application-Level Observations”, ACM Transactions on Internet Technology, Vol. 5, No. 3, August 2005

[3] Xiaohui Long & Torsten Suel, “Three-Level Caching for Efficient Query Processing in Large Web Search Engines”, Proceedings of the international conference on World Wide Web, 2006.

[4] I. Stoica, R. Morris, D.R. Karger, M.F. Kaashoek, H. Balakrishnan, “Chord: a scalable peer-to-peer lookup service for Internet applications". SIGCOMM, 2001, pp. 149-160.

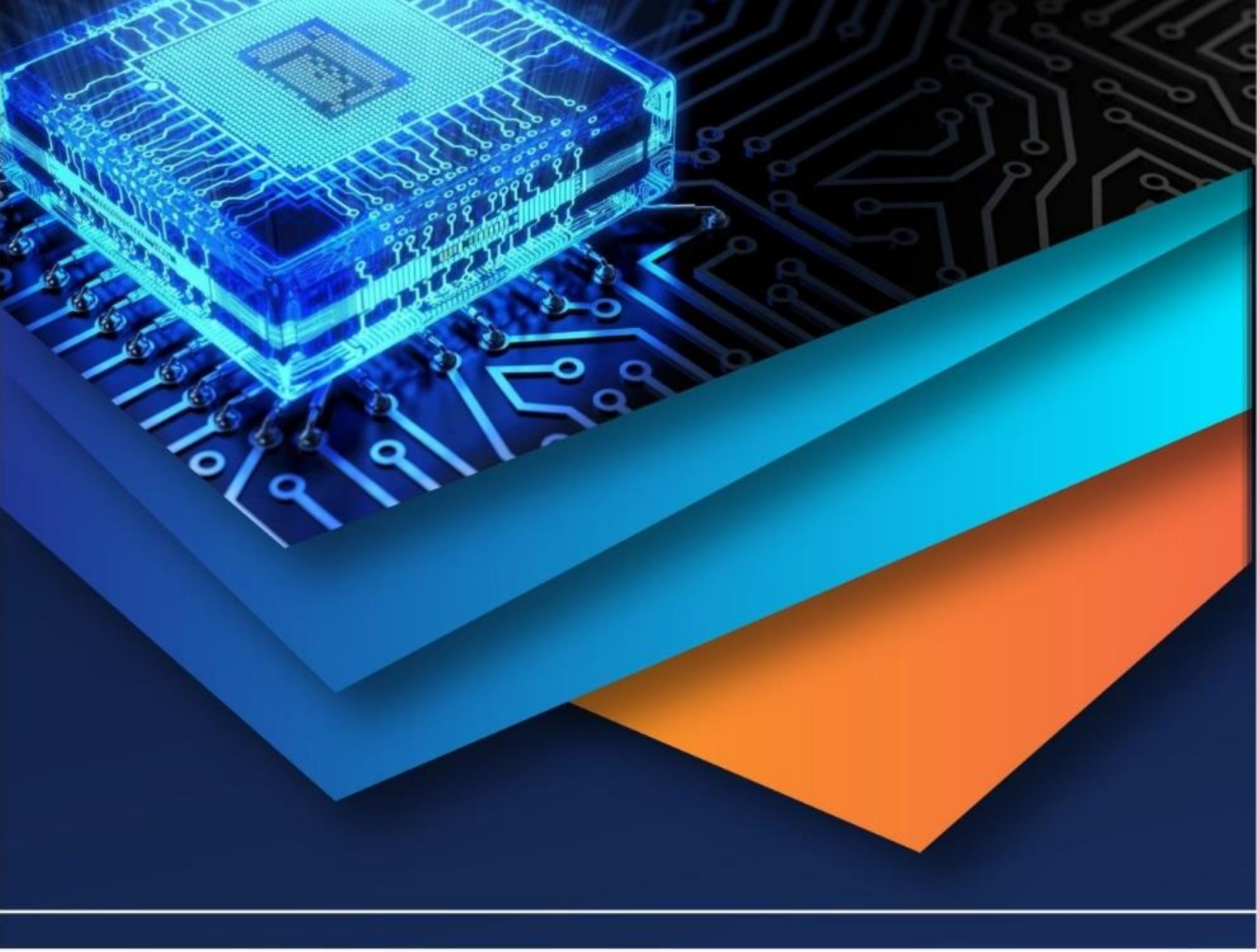
[5] S. Iyer, A.I.T. Rowstron, P. Druschel, Squirrel: a decentralized peer-to-peer web cache, PODC, 2002, pp. 213-222.

[6] N.H. Erika Rosas, M. Marin, “Two-level result caching for web search queries on structured p2p networks,” Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems, 2012, pp. 221-228.

[7] Gan, Quingging and Torsten Suel, “Improved techniques for result caching in web search engines,” Proceedings of the international conference on World Wide Web, ACM, 2009

[8] Shudong Jin and Azer Bestavros, “GreedyDual* Web Caching Algorithm,” NSF research, 2006.

[9] [9] J. Zhao, P. Zhang, G. Cao, and C. R. Das, “Cooperative caching in wireless p2p networks: Design, implementation, and evaluation", IEEE Transactions on Parallel and Distributed Systems, Vol 2, Sept 2010, pp.229-241.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY