ISSN:0975-9646

Rajashree Dongare et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1556-1559

Enhancing the Query Processing in Unstructured Peer To Peer Network

Rajashree Dongare[#1], Prof.Rachana Satao[*2],

[#] *Department of Computer Engineering, SKNCOE, Pune-411041,India*

Abstract -Normally in daily life we mostly use unstructured peer to peer network in which response latency and traffic cost of query fetching are two main performance factors. Controlled flooding query resource query algorithm is widely used in unstructured peer to peer network which locate the item which has minimum message cost, But it also require a large amount of time. So, here we have proposed a novel algorithm which calculates optimal combination of TTL value and set of neighbors by using history table to control scope of next query. Due to which any data which is previously searched can be fetch easily in next iteration and if we use web services for peers then we can achieve a good performance than normally we getting now.

Keywords*——** ***Peer to Peer network,Webservices, Selective Dynamic Query, Query Algorithm, TTL value, Backup Table.

I. INTRODUCTION

Peer to peer network is a type of network in which each node is act suppliers and consumers of resources. Basically topologies of peer to peer system are of three types: centralized, decentralized but structured and decentralized and unstructured. Here I am going to use unstructured topology because1) this type of network is simple to implement 2) having little overhead. Here we basically focus on resource query problem in unstructured peer to peer network. And wireless network is self organizing structure. So here we are using it.

P2P overlay networks, running at the application layer, perform scheduling and routing without any knowledge of the underlying physical networks . Various peer-to-peer systems have became the most popular Internet applications and a major portion of the Internet traffic is attributed to them. Another approach towards peer discovery is query flooding, which is used by newer applications such as Gnutella.The premise here is that instead of relying on a central directory server, a peer would directly broadcast a query to the network, and whomever has the desired resource would respond. Notably, in this approach, there is no central point of failure.

However, flooding the network has bandwidth usage considerations that could as well lead to an unintended self distributed. A variant of the query flooding approach is to select certain, high-availability and high-capacity nodes, as super nodes. These super nodes are given the task of indexing peers within its own domain and answering and creating queries from and to other super nodes. This approach reduces bandwidth usages by a large margin, but it does not really remove the inherent problems with query flooding.

Network is represented by G= (V, E) where we need to find result of query with minimum traffic cost and response latency. Where traffic cost is number of messages to get the result of query and response latency in the time period required to get the required result of query. We are using Database Backup concept For Using Security Purpose. If Connections are failed in main database then Results will be stored in Backup Database

To deal with the searching query in unstructured peer to peer network we use DQ method. It tries to minimizes the number of nodes that is require to visit in a network. Here the query initiator start finding the required file by communicating with neighbours which having small number of TTL value. SDQ is 1) well planned—every query round always tries to finish the query in a small TTL value, so that the inquiry node can limit the range of flooding and the chance of overshooting, and reduce perspective response latency; 2) greedy—in each iteration, the inquiry node propagates the query packets to the calculated subset of neighbours with the calculated TTL value, expecting to obtain the required number of results via these neighbours at one time; and 3) safe—with relatively low integer TTL values, SDQ can be more aggressive on neighbour selection Take TTL ¼ 3=nTTL ¼ 2 as an example:

selecting one more neighbour would incur just hundreds of visited peers and a few overshooting results; in DQ+, an aggressive TTL value may bother thousands of peers.

SDQ is in fact an optimization process with two objectives: minimizing both response latency and traffic cost. If in each iterative round TTL value is selected wisely and the query

packets are propagated to the right number of neighbors, it can be expected that within only one or two iterations, there would be enough returned results and the cost and atency could be minimized.This is the intuition behind our approach: the inquiry node always tries to exploit the degree heterogeneity of all residual neighbors. As mentioned above, DQ(f)+ also avoids TTL value rounding, while users are unlikely to have incentives to upgrade. The SDQ algorithm does not require any upgrade in other peers except the inquiry node itself; hence, the flag data for transition are not necessary.

While requesting the file it first search into backup database, if it is present over there then it will return it otherwise it will calculate TTL value and group of neighbor.

II. LITERATURE SURVEY

And second one is random walk-based algorithm. In which the node which wants the result will send a query packet in random manner until it finally hits the target. This algorithm can reduce the network traffic but it requires longer search latency in unstructured peer to peer network[3].

In Towards minimum traffic cost and minimum response latency: A novel dynamic query protocol in unstructured p2p networks they have calculated the TTL value of node and after that at each iteration TTL value is set to be as larger value and after that query packet is propagated to a bigger number of peers which causes less iteration. But this method is greedy i.e. in each iteration query node sends a query packet to new neighbors hoping to obtaining all required number of result [4].

In exploiting dynamic querying like flooding techniques in unstructured peer-to-peer networks [5] they have find the flow of dynamic query which is less attractive in search latency. They have done only little modification which improves the performance of previously used methods.

In Enhancing and Analyzing Search performance in Unstructured Peer to Peer Networks Using Enhanced Guided search protocol [7] they have given Without stringent constraints over the network topology the unstructured P2P networks can be constructed very efficiently and are therefore considered suitable to the Internet environment. But, the random search strategies adopted by these networks usually perform poorly with a large network size. To enhancing the search performance in unstructured P2P networks through exploiting users' common interest patterns are captured within a probability-theoretic framework termed the user interest model (UIM). A search protocol and a routing table updating protocol are used to obtaining the search process through self organizing the P2P network into a small world. In this paper both theoretical and experimental analyses are conducted and demonstrated the effectiveness and efficiency of the approach.

But it cnt be used in long distance devices. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan suggested Structured Peer-to-Peer (P2P) systems like Chord keep association of resource identifiers to nodes using a Distributed Hash Table (DHT), which allows to locate the node responsible for the resource with a given Id (or key) with logarithmic performance bounds[8].

Chi-Jen Wu, Kai-Hsiang Yang and Jan-Ming Ho focuds on the free-rider problem in the DQ like search algorithm, and propose a new search algorithm, called “AntSearch”, to reduce the redundant messages during a query flooding. In AntSearch, each peer maintains a pheromone value to present its hit rate of previous received queries, and records a list of pheromone values of its immediate neighbors. Based on these pheromone values, the AntSearch can flood a query only to those peers which are not likely to be free-riders[11].

.

III. PROBLEM STATEMENT

As everyone want any reply in a short amount of time so there is main problem regarding the time required to request and response i.e. we are basically interested in finding the data with minimum traffic cost and minimum response latency where traffic cost is nothing but number of messages required to obtaining the desired data and response latency is the time period required to obtaining the desired file. Generally speaking, it is difficult solve integer optimization problems because of their inherent combinatorial complexity. so there are some problem that cn be generated:-

1) Repeatedly searching over the network which is time consuming. For that we can use history table for storing the behaviour of peers in the network ,so that we can get required result in short amount of time.

2) As mobile nodes are movable so the IP address are randomly allocated so that will be the another issue while searching desired data as we are storing the total history of previous searched data with particular IP address of node.so that factor also can be improved by using getting MAC address of node.

IV. MATHEMATICAL MODEL

1. Let U be user or Enquiry node
U = {u1, u2, u3,…., un}

2. Let TTL value indicates the hops from the Farthest reached node to the inquiry node.
nTTL value (nTTL= TTL – 1)

3. Let The average degree of the network be D

4. Horizon H refers to the expected number of queried peers in a query round.
H = (d -1) ∑ nTXTL-1 (D-1) i

5. nTTL Value can be computed as
nTTL ≈ log(D-1)H(D- 2)/d – 1

6. let Rc denote the number of results that already collected.

Let Hes be calculated by the summation of the estimation of all finished rounds
Pes = Rc/Hes

B. Phase 1

1) Pseudo Codes of An Iterative Phase Algorithm

1: Rl --> results need - results received {results number remains to be retrieved}

2: Hes -->horizon estimated {estimated number of touched nodes}

3: Dl -->degree remain {total degrees of available neighbors}

4: Hne Estimation (Rl;Hes) {estimation of next horizon}

5: Dne,nTTL --> NextQueryTTL(Hne,Dl) {calculate proper nTTL and required degree for next query.}

6. SelectQuerySet(Dne,nTTL) {select a proper set of neighbors}

C. Phase 2

1) Selecting Next TTL

Given the estimated horizon of the next round, an optimal nTTL value can then be derived

$d \approx H(D-2)/(D-1)nTTL + 1$

2) Calculating Next Query Set

1: Initialization
2: ListHead ={index to available neighbor list}
3: Dm= degree of head
4: Nberlist=neighbour list

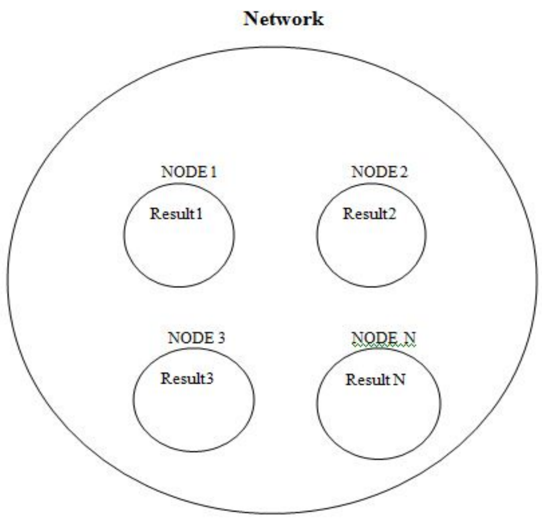

Fig. 1 Venn diagram

As here is we can say different peer as a set in a diagram. And each node contains the result. The querying node will search that result in neighbours node if it will find it will stop searching, but if it will not get expected result then it will start it in next query.

V. PRAPOSED ARCHITECTURE

In proposed architecture enquiry node will fire a query and at the same time it will calculate the TTL value of neighbours and group of neighbours. And by observing this it will select the neighbour node having less TTL value and optimal solution for neighbours.

That is this architecture will work on two phases.1)Probe Phase:- This is same as DQ.2)Iterative Phase:- Based on the estimated horizon of next query and the total residual degrees of all unused neighbours, SDQ calculates a proper integer nTTL value and the number of required degree; after that, a subset of the neighbours are chosen according to the number of required degrees; query packets with this TTL value are then propagated via these neighbours. The iteration process stops if enough results are returned; other-wise, a new query round is initiated.

Then for calculating TTL:-) Selecting Next node with small TTL the most multiobjective optimization problem is hard to find an optimal solution. Fortunately, we can exploit the integrity of the TTLvalues.The number of all the residual neighbour's degrees Dl is the key here. For each possible nTTLvalue we calculate the node In Fig.2 we have shown our proposed architecture, in which user will send a query to neighbour. If that node will have desired file it will reply otherwise it will send that query to other node

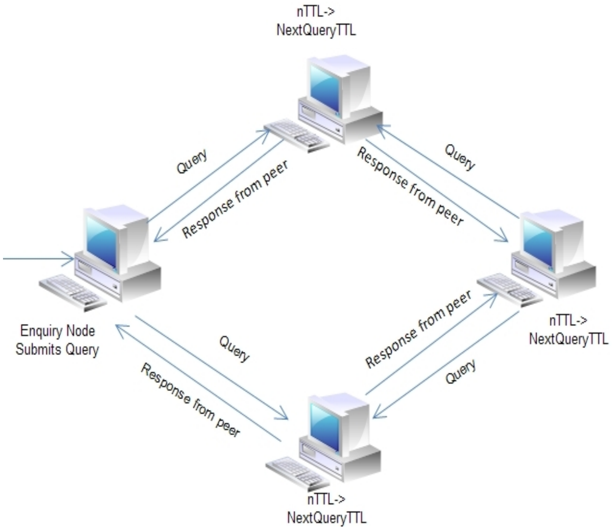

Fig. 2 Architecture of Proposed Peer to Peer network

For calculating next query set, Selecting the optimal subset of residual neighbours to reach can be solved by mathematical programming. Even with a smaller confidence level than DQ+, the popularity and horizon estimations are still conservative in SDQ. Here, in neighbours' selection we can deliberately introduce a little remedy to achieve balance .

1) In short after getting the desired result it will obtain the path for it and after that it will store that path in a history table so that in next Iteration if it want the same file then instead of calculating all the parameters it will simply lookup the table and follow the path for searching.
2) And in case of IP address if it get changed at that time we can follow the path by accessing the physical address of node. So that we will be not depend on IP address of node.
3) And if file is get deleted from source node and after that if file is get searched then it will not be downloadable from source node then its entry from history table will get deleted automatically.

VI. METHODOLOGY

If query which user wants to search is already present in History table then it directly picked up otherwise it calculate optimal combination of an integer TTL value and set of neighbor for next query round. As for each request we determine nTTL value for next query around and we calculate degree of next node .TTL value if the Time to Live for a particular request that is to say If the request doesn't get reply within a given time then that request is discarded by a particular peer.

For the communication and finding the desired data webservices are used. The characterization of a P2P network is the peer-to-peer communications model. Each functional unit in the network is behaviorally similar. There is no fixed role for a peer as there would be with the client/server distribution paradigm. Peers may do exactly the same thing or there may be transienst role of assignment.

And the characterization of the Web service operation is the classic client/ server model. Software that fulfills the role of the client will contact the datastore to discover the server location and can then contact the server. The mechanism to ensure that the client and server can speak intelligibly to each other, or interoperate, is enforced by well-known standards. This is a straightforward approach to distribution with the advantage that clients are coupled to the servers only by a software contract.

So if webservices used in P2P network then any node can contact with any other node with help of a software contract only .For ex.One program contacts another program directly and they can talk.Like may I got your IP address and port number?then another node simply send its IP address and other services whatever it have with it.

VII. Conclusion

Hence by using the backup table we can achieve any file in short amount of time due to which traffic cost and response latency can be reduce.

VIII. Result And Observation

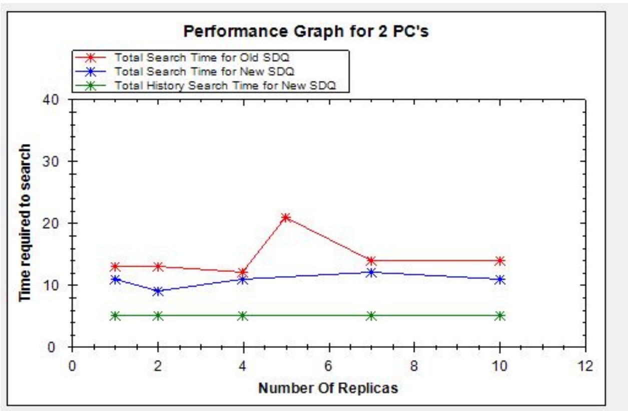

Fig 3. Performance graph of proposed SDQ for 2 PC

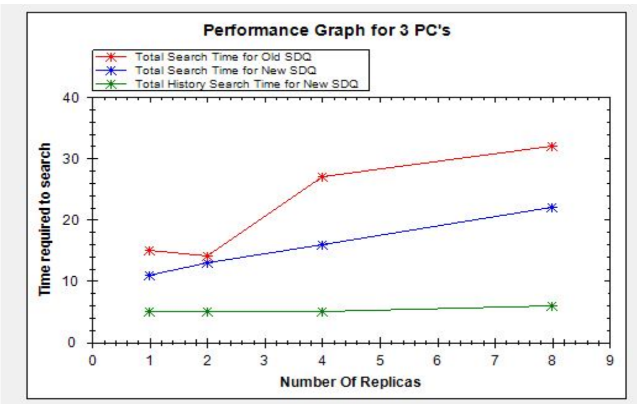

Fig 4. . Performance graph of proposed SDQ for 3 PC

References

[1] ChenTian,Hongbo Jiang,Xue Liu,Wenyu Liu"Revisiting Dynamic Query Protocol in Unstructured Peer-to-Peer Networks",Jan 2012.

[2] N. Chang and M. Liu, "Revisiting the TTL-Based Controlled Flooding Search: Optimality and Randomization," Proc. ACM MobiCom, 2004.

[3] Q.Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," Proc. Int'l Conf. Supercomputing, 2002.

[4] C. Tian, H. Jiang, X. Liu, W. Liu, and Y. Wang, "Towards Minimum Traffic Cost and Minimum Response Latency: A Novel Dynamic Query Protocol in Unstructured P2P Networks," Proc. IEEE 37th Int'l Conf. Parallel Processing (ICPP), 2008.

[5] H. Jiang and S. Jin, "Exploiting Dynamic Querying Like Flooding Techniques in Unstructured Peer-to-Peer Networks," Proc. IEEE 13th Int'l Conf. Network Protocols (ICNP), 2005.

[6] Mengkun Yang and Zongming Fei,"Assigning Identification to Nodes in Unstructured Peer-To-Peer Networks:A Novel Approach to improveing search Efficiency"Dec2007.

[7] Anusuya.R1, Dr.Kavitha.V2, Mrs. Golden Julie.E3," Enhancing and Analyzing Search performance in Unstructured Peer to Peer Networks Using Enhanced Guided search protocol" June 2010.

[8] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, 2001.

[9] Domenico Talia and Paolo Trunfio," Dynamic Querying in Structured Peer-to-Peer Networks", Proc. of 19th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2008).

[10] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Scott Shenker, "Making Gnutella-like P2P Systems Scalable ," In Proceedings of ACM SIGCOMM 2003.

[11] Chi-Jen Wu, Kai-Hsiang Yang and Jan-Ming Ho," AntSearch: An Ant Search Algorithm in Unstructured Peer-to-Peer Networks",Anusuya.R1, Dr.Kavitha.V2, Mrs. Golden Julie.E3 "Enhancing and Analyzing Search performance in UnstructuredPeer to Peer Networks Using Enhanced Guided search protocol".