ISSN: 2320-1363

Distributed Sharing of Resource in peer -peer (P2P) Network

Dr. Bhaludra Raveendranadh Singh

Principal, Visvesvaraya College of Engineering & Technology, Ibrahimpatnam, Dist: R.R, Hyd, AP, India

Abstract

The networking concept starts from connecting two devices may be computer, laptop, mobile etc. In this paper; Peer-peer network is most common network which we have already come across in our networking mechanism, but still there is a great lot of demand in making the technology more advance. Sharing the resource in a peer – peer network in a proper manner i.e. in a structured manner which in turn we call it as distributed Sharing of resource over the network. The utility and efficiency of the network can be achieved through mainly two concepts (i) Collaborative mechanism, (ii) Dedicated network registration. Although there are many researches paper has published to optimize such a network with high effective techniques and routing schemes. In this paper, we likely to stress upon the collaborative mechanism , as of these days technology changing and enhancement is common factor, but still this paper will be effective in terms of high end cloud computing. Collaborating in a network should and must be authenticated; otherwise it will give rise to security issue. To get such privilege we have given authenticated at security socket layer. The properties of this paper will implement the cost effective and efficient flow of data in a secured a layer and in an optimized manner.

Index Terms- Peer-Peer network, shortest path Algorithm, Fast algorithm

1. Introduction

Networking is a field, which make this world as global village, thanks to technology and thanks to such a great brain who have dedicated their time to make such a "second click world ".The era of computer science started a long ago, but these shows the tremendous change in technology, providing a great lot facility to easy access to human being. Despite their popularity, most of the current unstructured P2P content distribution systems suffer from certain serious limitations. One such limitation is their simple, on demand mechanism for content discovery. Peers in these systems discover data items by circulating queries within the overlay network. A peer receiving a query responds back to the initiating node if it has any matching content. Upon processing a query, the recipient node removes it from its local buffers1. Thus, a query expires after it completes its circulation within the network. In other words, the network forgets the queries once they have completed their circulation. Some systems cache recently received queries. But it is done in an ad hoc fashion and for very short durations purposes, we call this the ad hoc query model, and we refer to the queries as ad hoc queries. The ad hoc query model suffers from two main shortcomings. First, an ad hoc query is only capable of searching and retrieving content that exists in the P2P network at the time the query was issued.Consider the scenario when a peer Pi issues a query at time Tb. Assume that the query completes its circulation in the network at time Tc. Clearly, this query cannot discover data items that were added after (Tc+_), where _ indicates the short duration of time for which the query might be cached at different peers. Further, the query will not reach a peer that joins the network after time Tc, and hence cannot discover matching content on the new peer. In this scenario, the only way for a peer to discover newly added data items would be to repeatedly issue the same query. This is not desirable, since it creates unnecessary traffic within the network. Second, P2P systems that are purely based on the ad hoc query model provide no support for peers to advertise or announce the data items they own to other interested peers. Advertisements are important for P2P applications where peers trade content.

2. Related Work

Several techniques exist in the literature for ranking items of a dataset of peer-peer network present a local algorithm that can be used for monitoring the entries in a certain percentile of the population. In their paper, the authors describe a majority voting algorithm, where each peer, Pi, has a real number bi, and a threshold $\tau > 0$ (the

ISSN: 2320-1363

same threshold at all peers). The goal is for the peers to collectively P determine whether i bi is above nτ where n is the number of peers in the network. This technique can be potentially used to find all the entries of the inner product matrix that belong to the pth percentile of the population. However, the major disadvantage is the communication complexity – a separate majority voting problem needs to be invoked for every inner product entry and thus the system will not scale well for large number of features. In the worst case, the communication complexity of the majority voting algorithm may become equal to the order of the size of the network. Distributed top-k monitoring by Babcock presents a way of monitoring the answers to continuous queries over data streams produced at physically distributed locations.

In this paper, we assume a central node and the top-k set is always determined by the central node. The coordinator node finds the answers to the top-k queries and distributes it to all the monitor agents. Along with it, the central node also distributes a set of constraints. These constraints allow a monitor node to validate if the current top-k set matches with what it finds from the local stream. If the validation results are true, nothing needs to be done. Otherwise, the monitor agent sends an alert to the coordinator node. The coordinator node re-computes the top-k set based on the current data distribution and sends out both the new top-k and new set of constraints to be validated by each monitor agent. Since the paper assumes that there is a central node, this technique is not directly applicable to many asynchronous large-scale networks such as Mobile ah-hoc networks, vehicular ad hoc networks and P2P networks which is the focus of this work. Fagin presents a way of combining query results derived from multiple systems. Often disparate databases and type of the query run on them return different types of results; Fagin's paper talks about combining them. It also proposes techniques to retrieve top-k elements from distributed databases. Our algorithm is applicable when there are a large number of nodes. Fagin's solution, when applied to our system, would require every peer to communicate resulting in a highly communication-intensive algorithm.

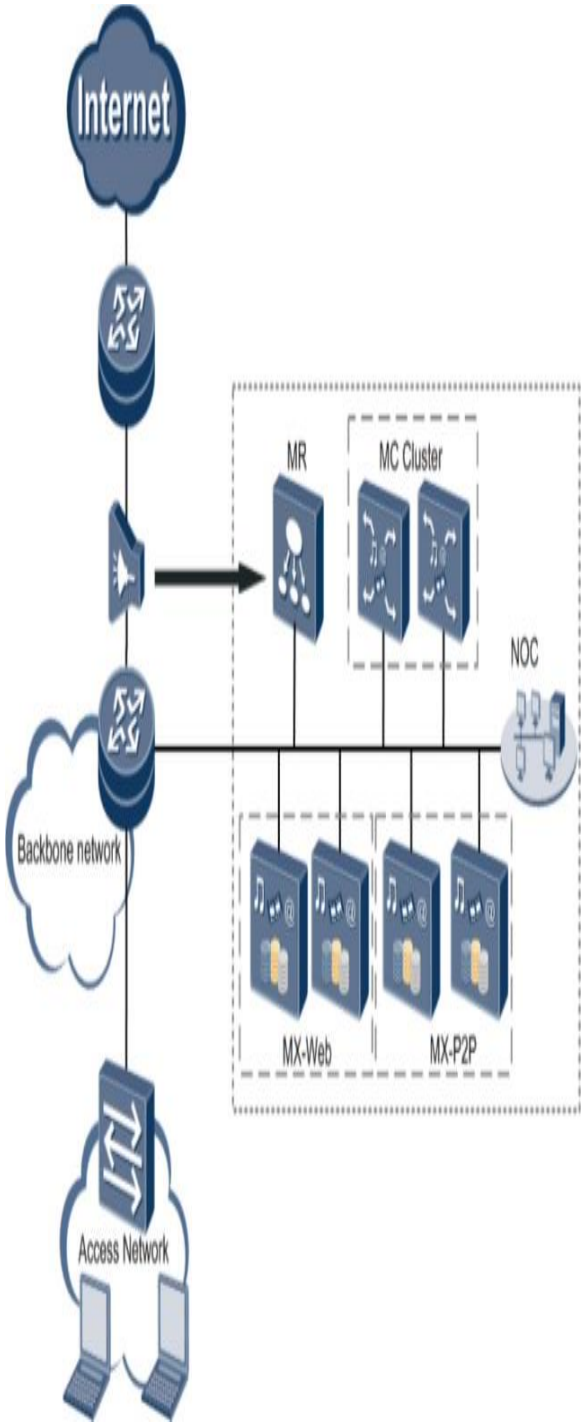

Fig.2.1 Showing Peer-Peer having Cluster

ISSN: 2320-1363

In the area of information retrieval, several techniques exist for top-k object identification. Balke et al. propose a super-peer approach for finding the top objects. The top queries are handled by the super peers and any other peer in the network can contact these super peers to get the answers to these queries. They also discuss ways to select these super-peers so that any peer can find its closest super peer efficiently. There are also techniques which explore the retrieval algorithms taking into account the relative rankings of objects. Many of these algorithms depend on gossip-based techniques for spreading the ranks of its objects. The major problems with gossip protocols are that they are slow (convergence can take a long time) and not very scalable due to global communication.

However, from our point of view, the most distinctive difference between Client/Server networking and Peer to-Peer networking is the concept of an entity acting as a Servant, which is used in Peer-to-Peer networks. Servant is an artificial word which is derived from the first syllable of the term server (îServ-ì) and the second syllable of the term client (ì-entî). Thus this term servant shall represent the capability of the nodes of a Peer-to-Peer network of acting at the same time as server as well as a client. This is completely different to Client/Server networks, within which the participating nodes can either act as a Server or act as a client but cannot embrace both capabilities.

3. Methods

In peer-peer network system, if we come across any of the system which is getting ready to communicate or in a communication, the mechanism might not be in strict fashion of topology or whatever the layout it follows. This may lead into perseverance notion among the network expert giving rise another misconception. Hence, this paper introduced such a mechanism, which can tackle such a context; the algorithm follows a route path may be of shortest path or some other related mechanism. Let's say you have a directed graph that represents airplane routes for some airline and you want to find the shortest path from one location to another. How can you do it so that it doesn't take all eternity? You have to use a priority queue that stores where you can go to from each point with its path length as the weight and you have two other arrays to store the predecessor for each location and the total distance for each element. You return that list of predecessors to trace the route, and the point of the total distance array is so that if you find a shorter path later on you can override what you currently have. The popping from the queue pops the least distant elements first of course. Here's what it looks like to clarify a bit.

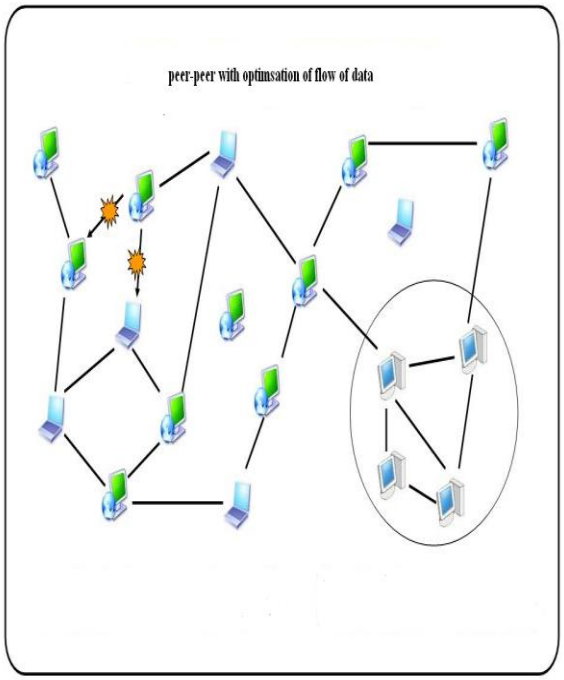

Fig.3.1 Peer-Peer (p2p) Network flow Diagram

A data processing system in which a computer performs a path search on a weighted graph, comprising: graph vertex memory in which vertex data on all of vertices in the graph is readably recorded; landmark memory in which values of the shortest path lengths d(Lv,Lw) are readably recorded, the values each found between any two vertices Lv and Lw selected as landmarks out of all of the vertices; adjacent-landmark memory in which values of the shortest path lengths d(v,Lw) are readably recorded, the values each found between any vertex v in the graph and each of one or a plurality of landmarks Lw adjacent to the vertex v; upper-limit calculation unit for calculating an upper limit of d(Lv,v)+d(Lv,Lw)+d(w,Lw) for any two vertices v and w in the graph, and for returning a value of the upper limit, by reading values from the graph vertex memory unit and the adjacent-landmark memory unit, the values being one or a plurality of landmarks Lv selected in association with the

ISSN: 2320-1363

vertex v as well as one or a plurality of landmarks Lw selected in association with the vertex w; lower-limit calculation unit for calculating a lower limit of d(Lv,Lw)−d(Lv,v)−d(w,Lw) for any two vertices v and w in the graph, and for returning a value of the lower limit, by reading values from the graph vertex memory unit and the adjacent-landmark memory unit, the value being one or a plurality of landmarks Lv selected in association with the vertex v as well as one or a plurality of landmarks Lw selected in association with the vertex w; and path-search unit for performing A* search by use of upper limits and lower limits of the shortest path lengths d(v,w) each found between any two vertices v and w, by receiving a starting point as well as a terminal in the graph and by calling up the upper-limit calculation unit as well as the lower-limit calculation unit, and thereby for outputting a result of the search, the result containing the shortest path between the starting point and the terminal.

Unstructured P2P networks-based pub-sub systems are relatively less explored. The main difficulty in designing pub-sub systems on unstructured P2P networks emerges from the fact that the topologies of most unstructured P2P networks have no relationship to the data (or other information) in individual peers. As a result, routing in these networks is, in general, independent of peers' contents (a few unstructured P2P networks use content-sensitive heuristics for routing). Most unstructured P2P networks-based pub-sub systems try to overcome these difficulties by carefully controlling the manner in which the topology of the overlay network evolves, and by adopting intricate indexing strategies for routing subscriptions and notifications. Unfortunately, these mechanisms are highly complex and they introduce significant overlay management overheads thereby limiting the scalability of the corresponding systems. As mentioned in the introduction, the Sub-2-Sub requires peers to be clustered on the basis of their subscriptions. The publisher of a data item is required to reach the cluster that exactly corresponds to the data item being published, and then start the dissemination process.

4. Conclusion

Technology is such a key word which is infinite the way you see or use it, hence of perseverance may take to great invention. In this paper some point like the algorithm like shortest path past and mining algorithm like fast algorithm to give pattern matching concept. By combing these two algorithms, we conclude such algorithm that can give easy, efficient and optimized solution to the field of small network; peer to peer network. Mechanisms that enable individual peers of unstructured P2P content sharing networks to register longstanding queries and receive notification when new matching items appear can significantly improve their utility and effectiveness. While the pub-sub paradigm can provide this capability, implementing pub-sub systems on unstructured overlays is often a very complex endeavor.

References

[1]GnutellaP2PNetwork. ww.gnutella.com.
[2] Kazaa P2P Network. www.kazaa.com.
[3] TIB/Rendezvous. White paper, 1999.
[4] S. Androutsellis-Theotokis and D. Spinellis. A Survey of Peerto-Peer Content Distribution Technologies. ACM Comput. Surv.,2004.
[5] B. Arai, G. Das, D. Gunopulos, and V. Kalogeraki. Approximating Aggregation Queries in Peer-to-Peer Networks. In Proceedings of the 22nd International Conference on Data ngineering (ICDE), 2006.
[6] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg. Contentbased Publish-Subscribe over Structured Overlay Networks. In Proceedings of ICDCS , 2005.
[7] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Ef_cient Multicast Protocol for Content- Based Publish-Subscribe Systems. In Proceedings of ICDCS 1999.
[8] T. Bu and D. F. Towsley. On Distinguishing between Internet Power Law Topology Generators. In INFOCOM, 2002.
[9] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event noti_cation service. ACM Transactions on Computer Systems, 19(3):332.383, 2001.
[10] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In Proceedings of ACM SIGCOMM 2003, 2003.
[11] J. Chen, L. Ramaswamy, and A. Meka. Message Diffusion in Unstructured Overlay Networks. In Proceedings of NCA, 2007.
[12] P. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/ Subscribe for RDF-based P2P Networks. In Proceedings of the 1st European Semantic Web Symposium, May 2004.
[13] P. A. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Designing
Semantic Pub-lish/Subscribe Networks using Super-Peers. Semantic Web and Peer-to- Peer.Springer Verlag, 2005.