DESIGN OF DISTRIBUTED, SCALABLE, TOLERANCE, SEMANTIC OVERLAY CREATION USING KNOWLEDGE BASED CLUSTERING

Ms. V.Sharmila
Associate Professor,
Department of Computer Science and Engineering,
KSR College of Engineering,
Namakkal.
Email: sachinsv06@gmail.com

Dr.G.Tholkappia Arasu
Principal,
Jayam College of Engineering and Technology,
Dharmapuri.
Email: tholsg@gmail.com

ABSTRACT

In a peer-to-peer (P2P) system, nodes typically connect to a small set of random nodes (their neighbors), and queries are propagated along these connections. Such query flooding tends to be very expensive. In this paper it is proposed that node connections be influenced by content, so that for example, nodes having many "Text" files will connect to other similar nodes. Thus, semantically related nodes form a Semantic Overlay Network (SON). Queries are routed to the appropriate SONs, increasing the chances that matching files will be found quickly, and reducing the search load on nodes that have unrelated content. It has been shown that SONs can significantly improve query performance while at the same time allowing users to decide what content to put in their computers and to whom to connect. In the work, the performance of semantic overlay has been analyzed using knowledge based clustering in terms of cluster formation time, searching time in the cluster and file transfer time.

Keywords- Semantic Overlay Network, peer-to-peer, distributed web search, content-based, similarity search.

1. INTRODUCTION

1.1 OVERVIEW OF OVERLAY NETWORK

An overlay network is a computer network which is built on top of another network.. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network. For example, many peer-to-peer networks are overlay networks because they run on top of the Internet.

The contribution of this paper is to propose a distributed and decentralized method for hierarchical SON construction that provides an efficient mechanism for search in unstructured P2P networks. Our strategy for creating SONs is based on clustering peers based on their content similarity.The current approach in web searching, i.e., using centralized search engines, rises issues that question their future applicability: 1) coverage and scalability, 2) freshness, and 3) information monopoly. Performing web search using a P2P architecture that consists of the actual web servers that has the potential to tackle those issues.P2P architectures can be classified into structured, like Chord [4] and CAN [6], and unstructured systems, like Gnutella [5]. Although structured P2P systems have recently received a lot of attention because they can guarantee retrieval of existing documents and provide upper bound on retrieval cost (in a better way than unstructured systems), they have a number of limitations that make them less suitable for the task of Internet scale web searching. For example, 1) peers indexing the most popular search terms will easily become bottlenecks, 2) when a peer joins the network each term that should be indexed has to be sent to the appropriate peer, 3) when a peer leaves, the terms it stores have to be reindexed, and 4) lack of support for efficient partial-match queries. These limitations do not occur in unstructured P2P systems. However, in order to make Internet-scale searching feasible, alternatives to the pure flooding-based search strategy have to be employed. Recently, the concept of Semantic Overlay Networks (SONs) has been proposed as an enhanced search mechanism. If SONs have been created, queries can be forwarded to only those sites that contain documents that satisfy the constraints of the query context, thus reducing the communication cost of the queries. The contribution of this paper is a decentralized and distributed method for semantic overlay network construction (DESENT), that provides an efficient mechanism for web search in unstructured P2P networks. Our strategy for creating SONs is based on clustering peers based on their content similarity (henceforth the word cluster will be used to refer to a SON and vice-versa).This is achieved by a recursive process that starts on the individual web sites. By applying clustering on the documents stored at each site, one or more feature vectors are created for each web site, more specifically one for each topic a site covers. Then representative peers,

each responsible for a number of peers in a zone are selected. These peers, henceforth called initiators,will collect the feature vectors from the members of the zone and use these as basis for the next level of clustering. This process is applied recursively, until we have a number of feature vectors covering all available documents.

2. OVERLAY NETWORK CREATION

SON generation process, assuming peers (for example web sites) storing documents and being connected in an unstructured P2P network. We refer to a zone as a set of peers in the same topological neighborhood. The initiator of a zone is the peer responsible for creating the zone and managing the zone's peers. A cluster is a set of peers that contain documents on the same topic(s). A cluster representative is a peer responsible for maintaining information about its cluster. Our approach is based on creating local zones of peers, forming semantic clusters based on local documents, and then merging zones and clusters recursively until global zones and clusters are obtained.

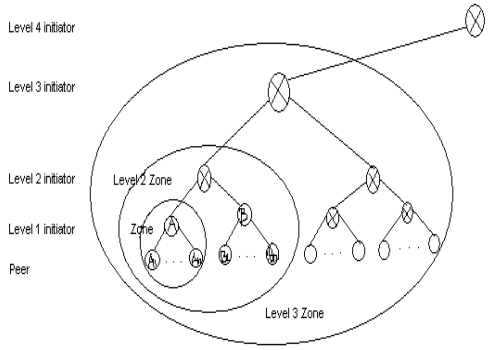


Figure 1 Hierarchy of zones and initiators.

3. SEMANTIC OVERLAY NETWORK

Semantic Overlay Network forms the semantically related nodes, and in peer-to-peer have the random nodes. SON groups the similar documents.

Semantic Overlay Networks (SONs), a flexible network organization that improves query performance while maintaining a high degree of node autonomy. With Semantic Overlay Networks (SONs), nodes with semantically similar content are "clustered" together. To illustrate, consider Fig.2 which shows eight nodes, A to H, connected by the solid lines. When using SONs, nodes connect to other nodes that have semantically similar content. For example, nodes A, B, and C all have "Rock" songs, so they establish connections among them. Similarly, nodes C, E, and F have "Rap" songs, so they cluster close to each other. Note that we do not mandate how connections are done inside a SON. For instance, in the Rap SON node C is not required to connect directly to F. Furthermore, nodes can belong to more than one SON (e.g., C belongs to the Rap and Rock SONs).

In addition to the simple partitioning illustrated as shown Fig.2. Also explore the use of content hierarchies, for example, the Rock SON is subdivided into "Soft Rock" and "Hard Rock."In a P2P system, the links between the nodes typically form a single overlay network. Advocate the creation of multiple overlay networks to improve search performance.

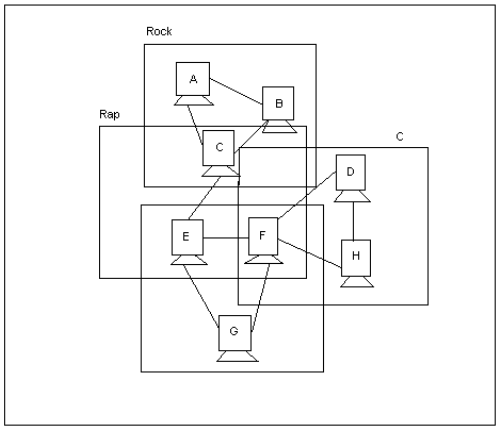


Figure 2 Semantic Overlay Network

4. DECENTRALIZED AND DISTRIBUTED OVERLAY CREATION

The peer clustering process is divided into 5 phases: 1) local clustering, 2) zone initiator selection, 3) zone creation, 4) intra-zone clustering, and 5) inter-zone clustering.

Phase 1: Local Clustering

Choosing feature vector for each cluster unit to segregate from other cluster unit. The feature vector we are choosing is small enough than the cluster units. These feature vectors are generated using feature extraction process.

Phase II: Initiator Selection

Assuming that one zone consist of collection of peers, from that zone we are randomly choosing a peer as an initiator. The chosen initiator is going to initiate and control all other cluster unit within the zone. The main objective of choosing the initiator is that uniformly spread out in the network, and it can be done using the MOD operator.

Phase III: Zone Creation

At the end of phase II all the initiator candidates are set to occupied and all other peer it is set to not occupied. Iterating those processes until the entire peer is occupied. A zone is split into two or more zones so that the resulting zones have equal size.

Phase IV: Intra-Zone Clustering

The global clustering starts with collecting feature vector from the peers. The local clustering creation involves several steps.

1. The initiator of each zone sends probe message to all other peers.
2. The peer which receives these messages will set the feature vector to the initiator of the zone.
3. The initiator performs clustering based on the received feature vector.
4. The initiator select representative peer for each cluster based on peer bandwidth, connectivity etc,
5. The result is kept at the initiator is a set of cluster description for each cluster.
6. Each of the representative peer is informed about the assignment and receives the copy of cluster description.

Phase V: Inter-Zone Clustering

Each initiator has identified a zone in its peer. Then we have to cerate overlay network that route the query to cluster in the remote zone. Applying the merging process repeatedly to form a super zone cluster. This activity involves much process that includes creating super zone by combining neighborhood zone and then neighboring super-zone is combined to a larger super zone etc.

5. FINAL ORGANIZATION

5.1. Hierarchy of peers

Starting with individual peers at the bottom layer, forming zones around the initiator peer which acts as a zone controller. Recursively neighboring zones form super-zones (see Fig. 1), finally ending up at a level where the top of the hierarchies have replicated the cluster information of the other initiators at that level. This is a forest of trees. The peers maintain the following information about the rest of the overlay network: 1) Each peer knows its initiator. 2) A level-1 initiator knows the peers in its zone as well as the level-2 initiator of the super-zone it is covered by. 3) A level i initiator (for i > 1) knows the identifiers of the level-(i-1) initiators of the zones that constitute the super-zone as well as the level-(i+1) initiator of the super-zone it is covered by. 4) Each initiator knows all cluster representatives in its zone.

5.2. Hierarchy of clusters:

Each peer is member of one or more clusters at the bottom level. Each cluster has one of its peers as representative. One or more clusters constitute a super-cluster, which again recursively form new super-clusters. At the top level a number of clusters exist. The peers store the following information about the cluster hierarchy: 1) Each peer knows the cluster(s) it is part of, and the representative peers of these clusters. 2) A representative also knows the identifiers of the peers in its cluster, as well as the identifier of the representative of the super cluster it belongs to. 3) A representative for a super-cluster knows the identifier of the representative at the layer above as well as the representatives of the layer below.

6. FAULTTOLERANCE AND RESILIENCE

The number of failures inevitably increases with the number of peers being involved. In a P2P network peer failures can be relatively frequent, and in order to ensure that no peer in the hierarchy becomes a single point of failure or a bottleneck this issue has to be handled efficiently. Our main approach is to use k-replication of important overlay network data, which is the hierarchy and cluster information. The replicated data is distributed on peers in a way that also distributes the tasks of the initiators over more peers. In the DESENT overlay network it suffices to replicate the overlay-related information stored at the initiators. This data is replicated at k - 1 other peer in the same zone. This replication is performed after the clustering process at level i and before the creation of the level-(i + 1) zone. During creation of the level-(i+1) zone the level-(i+1) is informed about the replica peers.

7. PEER JOIN

A peer PJ that joins the P2P network first establishes connection to one or more P2P neighbors as part of the basic P2P bootstrapping protocol (the actual protocol depends on the variant of unstructured P2P network, possible techniques include use of “known peers” as well as multicasting). These neighbors provide PJ with their zone initiators. Through one of these zone initiators PJ is able to reach one of the top-level nodes in the zone hierarchy and through a search downwards find the most appropriate lowest-level cluster which PJ will then subsequently join. Note that no reclustering will be performed, so after a while a cluster description might not be accurate. However, the global clustering process is performed at regular intervals and will then create a new clustering that reflects also the contents of new nodes (as well as new documents that have changed individual peer’s feature vectors). This strategy considerably reduces the maintenance cost in terms of communication bandwidth compared with incremental reclustering, and also it avoids the significant computational cost that could be the result of continuous reclustering.

8. PEER LEAVE

A peer can leave the network in two ways: 1) graceful departure where it notifies other relevant peers in the overlay network, or 2) leaving without notice, i.e., similar to a peer failure. In our system, both cases are treated similar to peer failure as described in detail in Section III-C. The only difference between the two is that in the case of a graceful departure a takeover message is sent to one of the peers containing the replica of its overlay network data, while in the latter case this process does not start until the failure is detected.

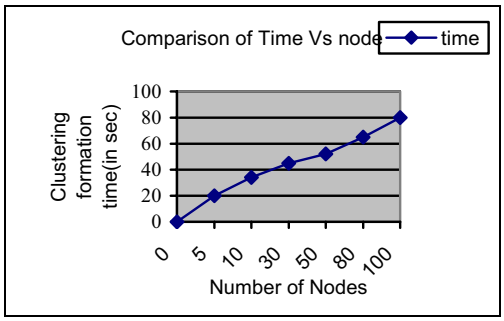
9. SEARCHING

When web search is performed, it is common that more than one document match the query. In our context, the aim is to direct a query Q to the cluster(s) that are most relevant for the query with respect to query terms QT . A query originates from one peer P, and it is continually expanded until satisfactory results, in terms of number and quality, have been generated. All results that are found as the query is propagated are returned to the query originator P. Query processing can terminate at any of the steps below, if the result is satisfactory. A query is distributed as described below. Q is sent to one of the top-level initiators (remember that each of the top-level initiators knows all top-level clusters).

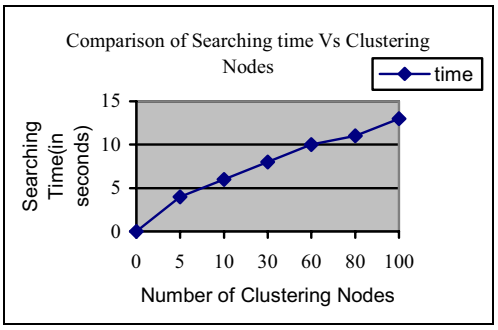
The most similar top-level cluster is determined, and Q is forwarded to its representative. Next, Q is routed down the cluster hierarchy, until the query is actually evaluated at the peers in a lowest-level cluster. The path is chosen based on highest similarity (sim(Q,Ci)) of the actual sub-clusters of a level-i cluster. If the number of results is insufficient, then backtracking is performed, in order to extend the query to more clusters.

In the experiments reported later in this paper the aim is to get as high recall as possible, and in this case the backtracking results in searching all forests that have sufficient similarity with the query. It should be noted that in practice a web search is satisfied by only finding the most relevant results, thus having a much lower cost.

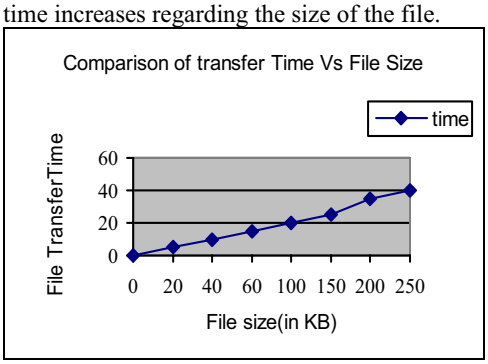
10. EXPRIMENTAL RESULT



In the above figure clustering formation time increases as the number of nodes increase. We have taken the above result with different run time.



In the above figure searching time increases proportional to the number of clustering nodes. In the above figure file transfer time depending on the size of the file. So file transfer time increases regarding the size of the file.



11. CONCLUSION

Examined the issues of supporting content-based searches in a distributed peer-to-peer information sharing system. Most existing Peer-to-Peer (P2P) systems support only title-based searches and are limited in functionality when compared to today's search engines. In this paper, we present the design of a distributed P2P information sharing system that supports semantic-based content searches of relevant documents.

Analyzed the performance of semantic overlay using knowledge based clustering in terms of cluster formation time, searching time in the cluster and file transfer time.

ACKNOWLEDGEMENT:

First of all we thank the almighty for giving us the knowledge and courage to complete the research work successfully. Express our gratitude to our respected Vice Chancellor Dr. Chelliah Thangaraj M.Tech., Ph.D for allowing us to do the research work internally. Also we acknowledge the support provided by TIFAC-CORE Network Engineering. (Department of Science and Technology, Government of India).

REFERENCES

1. A. Crespo and H. Garcia-Molina, "Semantic Overlay Networks for P2P Systems," Stanford University, Tech. Rep., 2002.
2. E. Cohen et al., "Associative search in peer to peer networks: Harnessing latent semantics." in INFOCOM, 2003.
3. K. Aberer, P. Cudr´e-Mauroux, M. Hauswirth, and T. V. Pelt, "Gridvine: Building Internet-Scale Semantic Overlay Networks," in Proceedings of International Conference on Semantic Web (ISWC'2004), 2004.
4. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proc. ACM SIGCOMM, 2001.
5. S. Michel, P. Triantafillou, and G. Weikum, "MINERVA Infinity: A Scalable Efficient Peer-to-Peer Search Engine," in Proceedings of Middleware'05, 2005.
6. P. Reynolds and A. Vahdat, "Efficient Peer-to-Peer Keyword Searching," in Proceedings of Middleware, 2003.
7. H. T. Shen, Y. Shu, and B. Yu, "Efficient semantic-based content search in P2P network," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 7, pp. 813–826, 2004.
8. P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos, "Towards High Performance Peer-to-Peer Content and Resource Sharing Systems," in Proceedings of CIDR'03, 2003.
9. B. Yang and H. Garcia-Molina, "Designing a Super Peer Network," in Proceed ings ICDE'03,2003.
10. C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid search schemes for unstructured peer-to-peer networks," in Proceedings of INFOCOM'05, 2005.
11. V. Cholvi, P. Felber, and E. W. Biersack, "Efficient search in unstructured peer-to-peer networks." in Proceedings of the Sixteenth Annual ACM Symposium on Parallel Algorithms, 2004.
12. C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in Proceedings of the ACM International Conference on Supercomputing (ICS), June 2002.
13. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In Proc. ACM SIGCOMM, August 2001.
14. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In Proceedings of SIGCOMM'03, 2003.
15. B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer systems," in Proceedings of the International Conference on Distributed Computing Systems (ICDCS).

BIOGRAPHY

V.Sharmila was born in Belukurichi, TamilNadu in 1981. She has received Master degree from Anna University Chennai and pursing her Ph.D in Anna University of Technology, Coimbatore. In the Academic year 2002-2004 she was a Lecturer in the department of Computer Science Engineering and 2005-2008 she was a senior Lecture and at present she is working as an Associate professor. Her research interest includes Data mining.

Dr.G.Tholkappia Arasu was born in Salem, Tamil Nadu in 1974. He has received Masters Degree from M.K University and Ph.D Anna University Chennai. In the Academic year 2009-2010 he was a Professor in the Department of Computer Science and Engineering and at present he is the Principal of Jayam College of Engineering and Technology, Dharmapuri.