A Novel algorithm for document search in unstructured Network

Kanthimathi S, Gayathri RU VA
imathi@outlook.com,ruva.gaya@gmail.com

Abstract: **Client-Server architecture is a network architecture in which each node on the network is either a client or a server. Each client is connected to a centrally located dedicated computer called server. The server must be fast and should have more storage capacity to contain all the data that needs to be shared to the clients. In peer-to-peer network, computers are connected individually such that there is no dedicated server. All the computers are equal, termed as peers. Each node on the network has the capability to share data and resources with other nodes. In another words there is no central authority that determines the network's resources sharing policy. Each user has the right to decide what he would or would not like to share. Each can act as both client and a server. The Domination set-based search algorithm is analyzed and the efficiency achieved is compared with the previous algorithms that existed earlier. More specifically, the various steps involved in searching, such as construction of a connected dominating set (CDS) and the associated reduction rules are studied. Performance results show that the Domination set-based search algorithm helps in achieving more efficiency, thus making it suitable for efficient searching of data in Peer-to-Peer networks in an effective manner.**

Keywords: **Peer-Peer, CDS marking, Searching.**

INTRODUCTION

The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a common exchange messages in a request-response messaging pattern: The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication. To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language separate hardware, but both client and server may reside in the same system. A server hosts one or more server programs which share their resources with clients.

Clients and servers and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. However, client-server model has its share of disadvantages.

Overloaded servers

When there are frequent simultaneous client requests, servers severely get overloaded, forming traffic congestion. But in a P2P network adding more nodes will increase its bandwidth since it's calculated as the sum of bandwidths of each node in the network.

Impact of centralized architecture

Since its centralized, if a critical server fails, client requests are not accomplished. Therefore client/server lacks robustness of a good P2P network (resources are distributed among many nodes).Peer-to-peer (P2P) is an alternative network model to that provided by traditional client-server architecture. P2P networks use a decentralised model in which each machine, referred to as a peer, functions as a client with its own layer of server functionality. A peer plays the role of a client and a server at the same time. That is, the peer can initiate requests to other peers, and at the same time respond to incoming requests from other peers on the network. It differs from the traditional client-server model where a client can only send requests to a server and then wait for the server's response.

OVERVIEW

The overview of this project is to develop a domination set-based search algorithm, which takes input from a node as file name and finds the node which contains the file in the Peer-to-Peer Network and return the file to the requested node.

Initially, the files from the standard database should be distributed among all the nodes in the peer-to-peer network and each node must maintain information about the files it contains. The distribution algorithm, which is the one of the modules module incorporated in our project is used to distribute the files among the various nodes of the peer-to-peer Network. The Algorithm is deployed in such a way that all the nodes in network will have the files and all files under each collection are distributed among the nodes in the peer-to-peer network.

For searching, any node can be a query node and sends a query file in the network and CDS is calculated using the marking process and two reduction rules. Searching is performed and node containing the requested file is obtained.

EXISTING SYSTEM

Gnutella is a large peer-to-peer network. It was the first decentralized peer-to-peer network of its kind, leading to other, later networks adopting the model. To envision how gnutella originally worked, imagine a large circle of users (called nodes), each of whom have gnutella client software. On initial start-up, the client software must bootstrap and find at least one other node. Various methods have been used for this, including a pre-existing address list of possibly working nodes shipped with the software, using updated web caches of known nodes (called Gnutella Web Caches), UDP host caches and,

rarely, even IRC. Once connected, the client requests a list of working addresses. The client tries to connect to the nodes it was shipped, as well as nodes it receives from other clients, until it reaches a certain count. It connects to only that many nodes, locally caching the addresses it has not yet tried, and discards the addresses it tried that were invalid.

Napster is another existing system, a name given to two music-focused online services. It was originally founded as a pioneering peer-to-peer file sharing Internet service that emphasized sharing audio files, typically music, encoded in MP3 format. The original company ran into legal difficulties over copyright infringement, ceased operations and was eventually acquired by Roxio. In its second incarnation Napster became an online music store until it was acquired by Rhapsody from Best Buy.

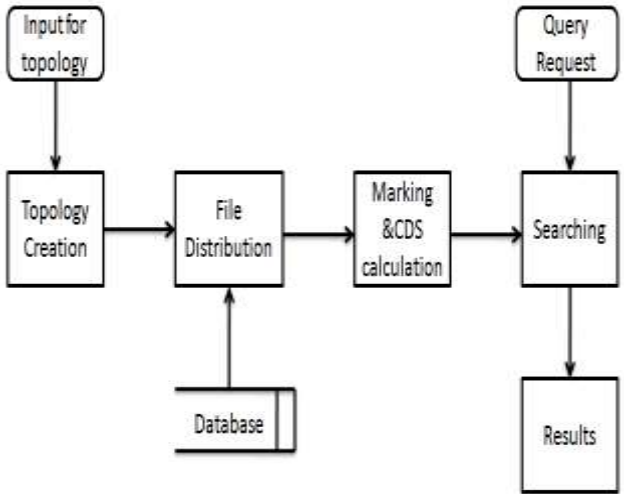
5.2 PROPOSED SYSTEM

The proposed system is fully decentralized and distributed network architecture in which individual nodes in the network (called "peers") act as both suppliers and consumers of resources, in contrast to the centralized client–server model where client nodes request access to resources provided by central servers.

It is the main representative of unstructured peer-to-peer network that do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other .P2P networks can be classified according to level of decentralization (centralized, decentralized or hybrid) or to the control over data location and network topology (structured or unstructured). Noticeably, we should solve wide range problems with P2P network maintaining and query processing despite the many advantages of P2P. The first significant problem is continuously joining and leaving P2P networks by nodes, known as "churn". System should have dynamic topology and be able to provide the same services regardless of the current network topology. The second is connected with first and is associated with query processing in dynamic network. The classical approach in unstructured P2P Gnutella-like networks is to use positive time-to-live (TTL) indicator to limit number of hops in a network: a query is transferred inside network until TTL expires. The primary goal is to design a overlay network that supports advanced searches. Such a network will have a variety of uses based on the enhanced search and retrieval that it offers. These range from file sharing to social networks to using semantic information to better manage networks. For example, our system will be able to support queries that find similar behavior (e.g., find all Denial-of-Service attacks that attacked a set of similar assets, or find all related causes of congestion). Data objects should be represented by a set of attribute values and/or metadata that describe the specific features and/or behavior that is desired to be retrieved. Attributes are decided according to the nature of the applications that are using that P2P system.

An efficient searching algorithm is the dominating-set-based peer-to-peer searching algorithm which helps in maximizing the return of searching results while keeping a low cost for both searching and creating/maintaining the connected-dominating-set (CDS) of the peer-to-peer network. A connected-dominating-set(CDS) of a peer-to-peer network is a connected subset of nodes of the network from which all nodes in the network can be reached

System Architecture



The goal is to develop a domination set-based search algorithm, which takes input from a node as file name and finds the node which contains the file on overlay network and return the file to the requested node. The files from the standard database should distributed among the nodes in the peer-to-peer networks and each node must maintain information about the files it contains. For searching any node can be a query node and sends a query file in the network and CDS is calculated using the marking process and two reduction rules. Searching is performed and node containing the requested file is obtained and requested file is given to the requested node

Distribution Algorithm

The distribution algorithm specified is used to distribute the files among the node of the peer-to-peer Network provided in the Standard TREC dataset. The dataset contains around 200000 files under 2500 collection names. The Algorithm is deployed in such a way that all the nodes in network will have the files and all files under each collection are distributed among the nodes in the peer-to-peer network.

The algorithm first finds the collection names and number of documents under each collection and then

selects a random node from an array which contains Ids of all nodes in the peer-to-peer network. Remove the node from the array for which the documents are assigned and check if the id array is empty and all the collections not are assigned, if true initialize the id array and repeat the process until all collections are assigned.

Algorithm 7.2.1: Distribution

Input:

Standard dataset with specified collection names and number of documents under each collection.

Output:

Documents distributed among the node.

Find the collection names and number of documents/collection

1. Select rand() node
2. Rand() node←number of documents in collection.
3. Remove() node ID from the array.
4. Check if IDArray[] is NULL and all collection not assigned, if YES
IDArray[] ←IDs //again.
5. Repeat 4 to 7 until all documents in all collections are assigned among the nodes.

Domination Set Based Search

A dominating-set-based peer-to-peer searching algorithm to maximize the return of searching results while keeping a low cost for both searching and creating/maintaining the connected-dominating-set (CDS) of the peer-to-peer network. A connected-dominating-set (CDS) of a peer-to-peer network is a connected subset of nodes of the network from which all nodes in the network can be reached. Finding a minimum CDS is NP-complete for most graphs. Marking process gives a simple and distributed algorithm for calculating CDS. In this project, we propose a peer-to-peer network searching algorithm using CDS generated by the marking process with some modification of the reduction rules 1 and 2 to maximize the searching results while keeping the cost of searching and maintaining the CDS low. This approach is based on random walk. However, the searching space is restricted to dominating nodes.

Marking &CDS Calculation

Specifically, the marking process is a localized algorithm in which hosts interact only with others in a restricted vicinity. Each host performs exceedingly simple tasks such as maintaining and propagating information markers. Collectively, these hosts achieve a desired global objective of finding a small CDS.

Two rules used 1-hop ranking, denoted as docs, of each node as the priority to break a tie. 1-hop ranking, docs, is defined as the total documentation number of node v plus the highest documentation number of a node's neighbor. We treat all types of documentation the same and will classify them in the future research. To get a unique total number of documentations, we can easily assign a unique node id. In case of a tie in the 1-hop ranking, node id is used to break a tie. The following algorithm Explains about the marking process and CDS calculation.

Marking and CDS calculation

First calculate
1-hop ranking =document the node has + documents of its neighbor node that has maximum documents
Marking process
Marker T is dominating node
Marker F represents non-dominating node
for(all Nodes){
If(Node's two neighbor not connected directly)
then
T←Node
else
F←Node }
CDS←all the nodes marked T
Reduction rules to get a CDS
rule 1:
if (N(A)N(B) in CDS && docs(A)<docs(B))
then
Remove A from CDS
rule 2:
If(Neighbor node A & B of node C are dominating &&
N(C)proper subset N(A) U N(B) in CDS && docs(C)<min(docs(A),docs(B))
then
Remove Node C from CDS

Searching

When a node S receives a request, S searches from its own database and returns the results to the requester if there is any documentation. If node S is the original request node and is not a dominating node (marked as F), it forwards the request to the dominating neighbor (marked as T) which has the highest 1-hop ranking among all of its dominating neighbors.

If node S is not the original requestor nor a dominating node, it will not send a query to any of its neighbors. If node S is a dominating node, it sends the request to the dominating neighbor with the highest 1-hop ranking among all of its dominating neighbors. Node S also sends the request to the non-dominating neighbor which has the highest 0-hop ranking among all of its neighbors(dominating neighbors and non-dominating neighbors) if there is one. Repeat process until either the maximum number of hops is reached or a visited node is reached again.

Searching

Input: Query Request
Output: file to the query node
Algorithm:
T is Dominating Set.
F is Non-Dominating Set.
1)Node S ← request, S searches its local database and returns result if present.
2) If(node S =original request node and marked as F)Then
dominating neighbor (marked as T) with highest 1-hop ranking ←request
3) If(node S=dominating node)Then

dominating neighbor (marked as T) with highest 1-hop ranking ←request
Non-dominating neighbor (marked as F) with highest documents ←request //if there is one

2) Repeat steps 1 to 3 until either the maximum number of hops is reached or a visited node is reached again.

IMPLEMENTATION

In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users, which it will work efficiently and effectively.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase, the most critical stage in achieving a successful new system and in giving the users confidence. That the new system will work be effective. The system can be implemented only after through testing is done and if it found to working according to the specification.

This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

CONCLUSION AND FUTURE SCOPE

Peer-to-peer (P2P) is an alternative network model to that provided by traditional client-server architecture. It is a decentralized model in which each machine, referred to as a peer, functions as a client with its own layer of server functionality.

Document Search in peer to peer networks using Domination Set Based Search Algorithm is the objective of this project. Initially, the topology is created based on the input provided for the network and the data (files) are to be distributed across the network. The conventional method of information (file) retrieval is, by using the concept of flooding. As it leads to wastage of bandwidth and also the efficiency is comparatively less, a highly efficient algorithm, which is the Domination set-based search algorithm has been incorporated into our project.

Before applying the Searching algorithm, files are distributed randomly among all the nodes in peer-to-peer network using an algorithm. Once the files are distributed, CDS(Connected-Dominating-Set) is calculated using marking process and two reduction rules as specified in the algorithm and for the requested query the searching algorithm is deployed and the node containing the file is obtained, thus acquiring the data which had to be retrieved in the beginning.

Thus the network can retrieve the file required, by searching only a selective set of nodes using the concept of CDS, rather than searching all the nodes in the flooding algorithm which is a tedious and a highly inefficient process. Hence, this algorithm helps in providing accurate results. The future enhancements can be based on the limitations of the searching protocol that has been designed in our project. The few limitations that are found in our project are:

- As the nodes are not initially checked whether they are malicious or not malicious in nature, the security remains an issue in the network.
- There may be some nodes in the network, which maybe down at a particular point of time. The data present in these nodes cannot be retrieved and hence, it is lost.

References

1. HariBalakrishnan, M. FransKaashoek, David Karger, Robert Morris, Ion Stoica∗ "Looking up data in P2P systems", Communications of ACM, vol.46, No.2,2003
2. Chunlin Yang, Jie Wu, "A Dominating-set-based routing in peer-to-peer networks" Proc of 2nd international workshop on Grid and Cooperative Computing Workshop,2003
3. N.Daswani, H.Garcia-Molina and B Yang, "open problems in data-sharing peer-to-peer systems", Proc of the 9th International Conference on Database Theory, 2003.
4. D.S.Milojicic, V. Kalogeraki, R.Lukoseetal.., "Peer-to-Peer computing", HP Lab technical report, HPL-2002-57, 2002
5. J.Liang, N.Naoumov, K.Ross, "The Index poisoning Attack on P2P file-sharing Systems", Infocom.
6. J.Risson, T.Moors, "Survey of Research Towards Robust Peer-to-Peer Networks:Search Methods" Technical Report, University of New South Wales, Sydney, Australia.2004
7. Xiuqi Li and Jie Wi "Searching techniques in peer-to-peer network", Department of Computer Science and Engineering

