IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Bootstrapping Non-Clustered Overlay Networks Using Cluster Formation Protocol

John Fernando.R[1] and Nanmohan.M[2]
[1,2]S.A Engineering College
Poonamalle-Avadi Road, Veeraraghavapuram, Thiruverkadu Post Chennai-600077

Abstract— In this project we introduce a Clustered-peer network which maintains a network topology that reflects peer sharing content interests .In this paper, we model the formation of such clustered overlays as a strategic game where nodes determine their cluster membership with the goal of improving the recall of their queries. Clustered-peers maintains records of files which are requested by peers with similar interests We show that, in general, decisions made independently by each node using only local information lead to overall cost-effective cluster configurations that are also dynamically adaptable to system updates such as churn and query or content changes.

***Keywords*— clustering, peer-to-peer, data sharing, overlay Networks,**

INTRODUCTION

Recently, there has been an explosion in the use of content Sharing applications such as those involving social Networking and peer-to-peer (p2p) file sharing. Measurements From the deployment of such large-scale systems have shown that the interactions among their participants or Nodes indicate the existence of groups or clusters of nodes Having similar content or interests. For example, in measurements of popular online social networks it was observed that the network structure is such that users form clusters Based on common interests, social affiliations, or the wish to exploit their shared content. The formation of implicit Groups centered on topics described by common keywords has also been observed in the blogosphere. Furthermore, the peer selection algorithm used in bit torrent was shown to lead to the formation of clusters of peers Having similar interests and upload capacities

Domain on Parallel and Distributed Systems:

A **distributed system** consists of multiple autonomous computers that communicate through a computer network. The computers interact with each other in order to achieve a common goal. A computer program that runs in a distributed system is called a **distributed program**, and **distributed programming** is the process of writing such programs .Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and may be seen as a loosely coupled form of parallel computing. In parallel computing, all processors have access to a shared memory. Shared memory can be used to exchange information between processors. In distributed computing, each processor has its own private memory (distributed memory). Information is

www.ijreat.org
Published by: PIONEER RESEARCH & DEVELOPMENT GROUP(www.prdg.org)

exchanged by passing messages between the processors.

Distributed database management systems:

A **distributed database** is a database in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers. Unlike parallel systems, in which the processors are tightly coupled and constitute a single database system, a distributed database system consists of loosely coupled sites that share no physical components. Collections of data (e.g. in a database) can be distributed across multiple physical locations. A distributed database can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks. The replication and distribution of databases improves database performance at end-user worksites.

Peer-to-Peer networks:

P2P is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged participants in the application. Each computer in the network is referred to as a node. The owner of each computer on a P2P network would set aside a portion of its resources - such as processing power, disk storage or network bandwidth -to be made directly available to other network participant, without the need for central coordination by servers or stable hosts. With this model, peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply (send), and clients consume (receive).

Overlay Network:

An **overlay network** is a Computer Network, which is built on the top of another network. Node 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, distributed system such as Cloud Computing, Peer-to-peer networks, and Client server applications are overlay networks because their nodes run on top of Internet. The Internet was built as an overlay upon the telephonic network.

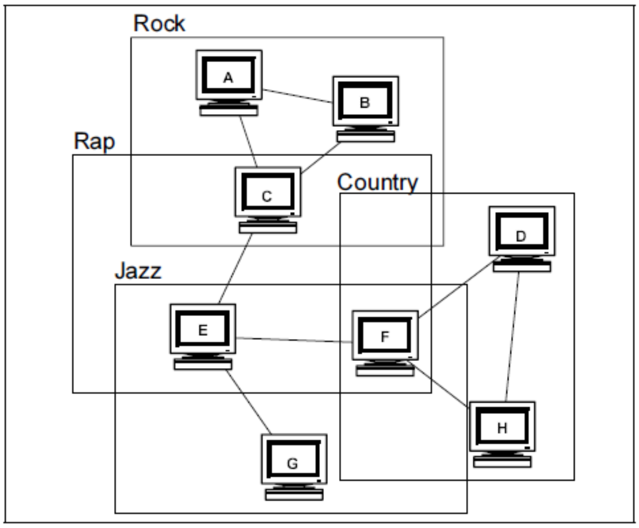

Fig1: Overlay networks

Cluster:

Cluster is a grouping, a group of loosely coupled computers that work together closely.
A cluster can also refer to several machines grouped together, all performing a similar function. For example, a cluster may consist of eight PCs, all connected via high-speed Ethernet, processing scientific data. This type of setup is often referred to as "parallel computing," since all the computers in the cluster are acting as one machine. Clusters are typically used for high-end processing, such as performing scientific calculations or decrypting
algorithm

IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

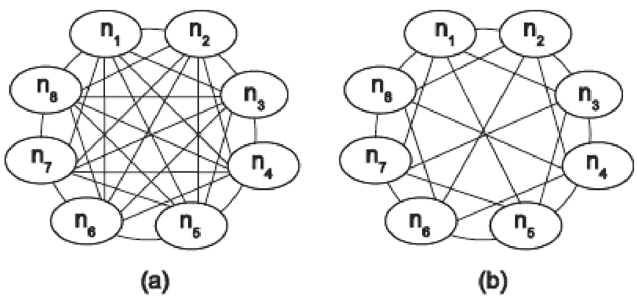

Fig2: Examples of cluster topologies

Existing System:

In this paper, Measurements from the deployment of such large-scale (content sharing applications) systems have shown that the interactions among their participants or nodes indicate the existence of groups or clusters of nodes having similar content or interests .The formation of implicit groups centered on topics described by common keywords has also been observed in the blogosphere. The main reason for the formation of such clusters is that clustered overlays enable their participants to find and exchange data relevant to their queries with less effort. For example, traces of popular p2p systems have indicated that nodes exhibit the property of interest-based locality, that is, if a node holds content satisfying some query of another node, then it most likely also maintains additional content of interest to this other node.

Disadvantage:

There has been a large body of research on the discovery and construction of clustered overlays. The larger the size of the cluster, the higher the cost of joining, leaving, and maintaining it.

Proposed System:

In this paper we propose the dynamic creation and adaptation of clustered overlays by taking a game-theoretic approach. We model the problem of cluster formation as a strategic game with nodes as the players. Each node plays by selecting which clusters to join. In our approach, we model clustered overlays and aim at increasing the recall of queries. We present an uncoordinated cluster formulation protocol that relies on local decisions made independently by each node based only on its partial view of the system.
A cluster configuration, where all nodes form a single cluster, the membership cost is maximized, while the recall loss is minimized, since for each node all results for its queries are located within its cluster. In contrast, the recall loss is maximized when each node forms a cluster of its own, while, in this case, the membership cost is minimized.

Advantage:

The cluster selection or strategy is determined individually by each node so as to minimize a utility function that depends on the cluster membership cost and on the cost of evaluating queries outside of the clusters.

Algorithm:

A Game theory Approach:

In Our Approach**,** We consider dynamic large-scale content sharing distributed systems. In such systems, it is not possible for a node to know and directly communicate with all other nodes in the system. Instead, each node establishes logical links with only a few other nodes. These logical links create a logical Overlay network on top of the physical one (e.g., the Internet).In this paper, we consider clustered overlays, where nodes with similar content or interests form groups, called clusters. The nodes inside each cluster are highly connected with each other to achieve an efficient intra cluster communication.

Uncoordinated Protocol

In our (basic) uncoordinated protocol, each node autonomously decides to play, i.e., revaluates its gain, after the evaluation of each of its local queries. Besides local queries, other nonlocal events (such as other nodes joining or leaving a cluster) may affect the gain of a node. For completeness, we also consider a variation, called uncoordinated protocol with monitoring, where a node re-evaluates its gain after any (local and nonlocal) event. Note that this protocol makes the unrealistic assumption that a node monitors the system continuously to detect potential updates that may affect its gain. The individual cost of each node depends on the recall of its queries and its cluster membership cost. Both quantities are estimated. To this end, we assume that each cluster has a unique identifier, cid, known by all its nodes, which is assigned based on node IPs and timestamps. For example, when the first node joins a cluster, its cid is formed by the IP of the node concatenated with a timestamp. When other nodes join the cluster, they are informed of its cid. When all nodes leave a cluster, its cid just becomes unused. Recycling Cluster ID is beyond the scope of this paper. Query results are annotated with the corresponding Cluster ID of the clusters that provide them. A node does not need to know all system Cluster ID, but gradually learns them, as its queries acquire results annotated with new Cluster ID. Based on the annotated query results, each node can monitor its recall with respect to the other clusters in the system and use it to evaluate its individual cost for the different configurations it needs to consider when it plays.

Cluster Reformulation Protocol

The cluster update protocol takes place periodically. The protocol proceeds in rounds where each round has two phases: in the first phase, relocation requests are gathered, and in the second phase, they are served. To avoid broadcasting messages among all peers, we assume that one peer per cluster acts as the cluster representative. The representatives of each cluster do not need to be the same in all rounds of the protocol. Note also that, it is not required that the cluster representatives of all clusters are known to each other. If the cluster representative forwards the request to a peer in the cluster, the peer can then propagate it to its representative. In the first phase of each round, each peer evaluates its gain factor (depending on the relocation policy applied) and sends this value to its cluster representative. Each cluster representative selects the peer with the highest gain value in its cluster and sends a relocation request to all other cluster representatives including its own cid, the cid of the cluster the peer wants to move to and the value of the gain. In the case where no peer needs to relocate, the representative sends just a message with its cid. When all representatives have received relocation requests from all other clusters in the system, the second phase of the protocol begins.

In the second phase of each round, each cluster representative sorts the relocation requests that it has received according to their gain value. To speed-up this phase, we try to avoid cycles due to groups of peers moving in loops among the same set of clusters. To achieve this, we enforce the following rule: if peer p 2 ci moves to cj, then ci is locked with direction *leave* and cj with direction *join*. In the same round, no more peers can join ci or leave cj . To enforce this condition, after the representatives have sorted the requests in decreasing order of gain value, the first relocation request in the list is granted. The two cluster representatives that are involved in the request communicate with each other to satisfy the request. Each cluster representative locks the two clusters, i.e., it removes all other requests in the list that involve either of them with a direction that violates the rule. The process continues by serving the next request in the ordered list. Note that cluster representatives can process their lists independently. After, all cluster representatives have processed their lists, the protocol proceeds to the next round. The protocol ends, when the peer representatives receive no further relocation requests.

Peer Clustering and Document Categories

The nodes of the system will be *logically* organized into a set of clusters. All nodes belonging to the same cluster will be able to either serve all the retrieval requests for documents contributed by all the nodes of that cluster (for example, in the case the nodes can store all documents), or find another node that can. The latter can be achieved by having each node, or a distinct set of super peer nodes, store cluster metadata, describing which documents are stored by which cluster nodes. In the following we will assume this latter design choice and discuss the type of metadata needed and its use. Alternatively, if pure P2P solutions are favored, the same goal can be achieved using routing indices2 at the cluster's nodes, routing requests for documents/categories to the proper cluster node(s). In the proposed architecture, clusters of nodes form storage collectives/repositories, and each cluster can store and thus serve requests for documents belonging to one or more document categories. Each category may belong to only one cluster. The nodes are assigned to clusters according to the categories of the documents they contribute. So, depending on how the categories are assigned to clusters, a node may belong to more than one cluster if it contributes documents associated with more than one category.

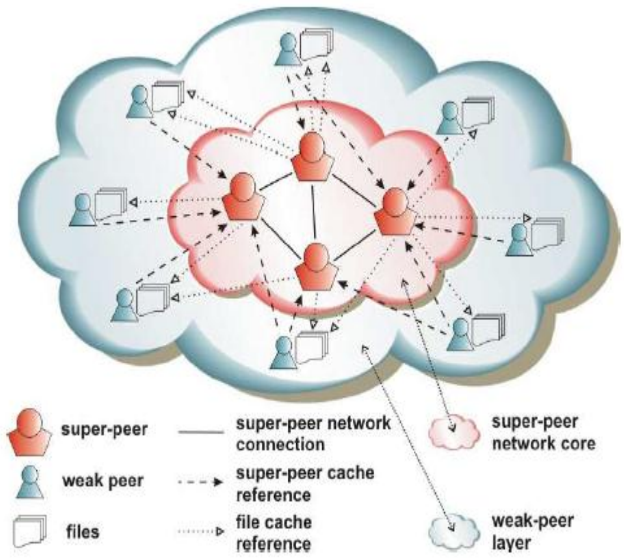

Fig3: A super peer network

Further Enhancement:

CLUSTER LEADER ELECTION:

In this module Leaders are elected periodically. Before the end of a period, nodes inform their cluster neighbours of their computing, storage, and bandwidth capabilities, while also forwarding relevant information received by other nodes. Thus, over time, all nodes of the cluster have a quite clear picture of the status of all nodes in the cluster, as far as processing, storage, and bandwidth capabilities are concerned. When the time has come for the system to enter the adaptation stage, the most powerful node in each cluster is chosen to be the leader of the cluster (thus a node can be the leader in more than one cluster). Note that this process may result in more than one peer believing to be the cluster leader (due to network partitioning, or when peers decide with incomplete information, for example). However, this poses no problem. During the adaptation stage, nodes probe their cluster leaders to assure they are alive. In the case of a leader failure, another node is selected to be the new leader. This can be the next more capable node, or a node close to the leader in the tree hierarchy (i.e., one that has most of the information needed for the adaptation and needs to send out only a small number of messages to acquire all remaining information). Furthermore, this guarantees that all nodes of a cluster know which node is currently their leader.

SUMMARY AND FUTURE WORK

In this paper, we have modeled the creation and maintenance of clustered overlays as a game. Nodes act as players that choose their strategy, i.e., which clusters to join, so as to minimize a utility function of the cluster membership cost and query recall. To cope with churn and query and content updates, nodes re-evaluate their strategies resulting in dynamic reclustering. There are many directions for future work. One is to consider altruistic nodes that aim at improving either the cost of other nodes or the overall system cost. Some preliminary results for the first case were reported in With regard to the model, possible extensions include adding to the cost function an explicit load-balance component as well as a component for the intercluster communication. Also, we plan to modify our protocol to efficiently handle multiple cluster membership without increasing its complexity. Finally, another direction is identifying possible connections between our game-based approach to clustering and traditional approaches based on distance measures.

REFERENCES

[1] E. Adar and B.A. Huberman, "Free Riding on Gnutella," First Monday, vol. 5, no. 10, 2000.
[2] N. Bansal, F. Chiang, N. Koudas, and F.W. Tompa, "Seeking Stable Clusters in the Blogosphere," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB), 2007. 596 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 4, APRIL 2012
.[3] M. Bawa, G. Manku, and P. Raghavan, "SETS: Search Enhanced by Topic Segmentation," Proc. 26th Ann. Int'l ACM Conf. Research and Development in Information Retrieval (SIGIR), 2003.
[4] A. Crespo and H. Garcia-Molina, "Semantic Overlay Networks for P2P Systems," technical report, Computer Science Dept., Stanford Univ., 2002.
[5] E.D. Demaine, M. Hajiaghayi, H. Mahini, and M. Zadimoghaddam, "The Price of Anarchy in Network Creation Games," Proc. 26th Ann. ACM Symp. Principles of Distributed Computing (PODC), 2007.
[6] A. Fabrikant, A. Luthra, E. Maneva, C.H. Papadimitriou, and S. Shenker, "On a Network Creation Game," Proc. 22nd Ann. Symp. Principles of Distributed Computing (PODC), 2003.
[7] P. Garbacki, D.H.J. Epema, and M. van Steen, "Optimizing Peer Relationships in a Super-Peer Network," Proc. 27th Int'l Conf. Distributed Computing Systems (ICDCS), 2007.
[8] S.B. Handurukande, A.-M. Kermarrec, F.L. Fessant, L. Massoulie´, and S. Patarin, "Peer Sharing Behaviour in the eDonkey Network and Implications for the Design of Server-Less File Sharing Systems," Proc. First ACM SIGOPS/EuroSys European Conf. Computer Systems (EuroSys), 2006.
[9] G. Koloniari and E. Pitoura, "A Game Theoretic Approach to the Formation of Clustered Overlay Networks," IEEE Trans. Parallel

and Distributed Systems, http://doi.ieeecomputersociety.org/ 10.1109/TPDS.2011.155.
[10] G. Koloniari and E. Pitoura, "A Recall-Based Cluster Formation Game in Peer-to-Peer Systems," Proc. Very Large Database Endowment (VLDB), 2009.
[11] P. Kuznetsov and S. Schmid, "Towards Network Games with Social Preferences," Proc. 17th Int'l Colloquium Structural Information and Comm. Complexity (SIROCCO), 2010.
[12] N. Laoutaris, G. Smaragdakis, A. Bestavros, and J.W. Byers, "Implications of Selfish Neighbor Selection in Overlay Networks," Proc. IEEE INFOCOM, 2007.
[13] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and Sharing Incentives in BitTorrent Systems," Proc. ACM Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS), 2007.
[14] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," Proc. Seventh ACM SIGCOMM Conf. Internet Measurement (IMC), 2007.
[15] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems," Proc. IEEE INFOCOM, 2003.
[16] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (SIGCOMM), 2001.
[17] E. Tardos and T. Wexler, "Network Formation Games," Algorithmic Game Theory, Cambridge Univ. Press, 2007.
[18] P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos, "Towards High Performance Peer-to-Peer Content and Resource Sharing Systems," Proc. First ACM SIGMOD/VLDB Conf. Innovative