Density-based Clustering for Similarity Search in a P2P Network

Mouna Kacimi, Kokou Yétongnon
University of Bourgogne
Laboratoire LE2I
Sciences et Techniques 21078 Dijon Cedex France
kacimi@khali.u-bourgogne.fr
kokou.yetongnon@u-bourgogne.fr

Abstract

P2P systems represent a large portion of the Internet traffic which makes the data discovery of great importance to the user and the broad Internet community. Hence, the power of a P2P system comes from its ability to provide an efficient search service. In this paper we address the problem of similarity search in a Hybrid Overlay P2P Network which organizes data and peers in a high dimensional feature space. Data and peers are described by a set of features and clustered using a density-based algorithm. We experimentally evaluate the effectiveness of the similarity-search using uniform and zipf data distribution.

1 Introduction

P2P systems have become in recent years one of the fastest growing and popular Internet-based systems. They consist in distributed scalable environments for resource and service sharing where peers can join or leave the network autonomously and frequently. P2P systems have raised several issues and challenges and have been the focus of many research studies. An issue central to P2P systems is information retrieval. In particular, the issue of approximate and complex queries is still an open problem. Therefore the need of efficient search algorithms assuring a high success rate for this type of queries in order to improve the performance of P2P systems. Several search techniques have been proposed in the literature, Some techniques aim to organize peers with similar interests in the same cluster to limit network flooding [1–3, 11, 14, 16, 17, 19, 20, 25, 27, 29]. Other techniques based on distributed hashing table DHT organize data in a key space for an efficient access and a complete lookup [10, 22, 26]. DHT techniques address appropriately the problem of exact queries but have limitations on approximate and complex queries.

Approximate search might be inherently expensive. First, the search space can grow exponentially with the number of peers and data. Second, peers containing all nearest neighbours to any query point cannot (1) belong to adjacent clusters in clustering approaches or (2) belong to adjacent zones in the key space of DHT systems. In this paper we present a density-based algorithm for a hybrid overlay network based on a peer organizing/data organizing combined scheme. This hybrid overlay network performs efficient approximate similarity search in a high dimensional feature space. The contributions of this paper are as follows:

- We introduce a density-based algorithm that performs similarity search for a hybrid overlay network (HON-P2P).
- We perform simulations to evaluate the control parameters of building clusters using uniform and zipf data distribution.
- We evaluate the performance of the similarity search using the density-based clustering algorithm and find that it achieves a high success rate.

The remainder of the paper is organized as follows. In the next section, we give an overview about the existing search techniques in P2P systems. Section 3 introduces the hybrid overlay network for organizing data and peers in a high-dimensional feature space. Section 4 presents the density-based clustering algorithm and the similarity search. Section 5 presents the simulations results. And finally section 6 concludes the paper.

2 Related Work

Previous work on P2P systems [4, 6, 7, 13, 18, 24] have focused to a large extent on what P2P architectures are, how to structure them using different characteristics, and how to define index structures and query routing schemes for organizing and retrieving peer contents. Generally, two

Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)
0-7695-2585-7/06 $20.00 © 2006 IEEE

main categories of P2P systems can be distinguished: unstructured systems and structured systems. Since the performance of a P2P system is determined by its efficiency to locate information, many search techniques have been proposed for both categories of P2P systems.

A main search technique in unstructured P2P systems is flooding [7] where each peer receiving a query broadcasts it to directly connected peers. However, Saroiu and al [23] noted that flooding technique is inefficient because it can overload the network. To solve this problem and limit the network traffic, the Time-To-Live(TTL) mechanism is used to restrict the distance a query can travel in the network. Another approach is to replace the flooding search with a random walker [5] which forward a query message to a randomly chosen neighbor at each step until the required object is found. These random techniques are efficient to locate popular data objects having replicate copies in a large amount of peers, thus the network traffic is decreased. Though, they might lead to excessive network bandwidth consumption, and remote or unpopular data objects may not be found due to the lookup horizon limit, typically imposed by TTL.

To address the flooding problems, the clustering technique involves the creation of links on top of unstructured P2P overlay networks to organize peers according to their common properties or interests. Several parameters can be used to cluster peers, such as network related information [1, 3], application needs [17, 29], peer characteristics [11, 16], and similarities between peers contents [2, 8, 14, 19, 20, 27]. Many methods are used to measure the similarity between peers. For example, Sripanidkulchai et al [14] take into account the query traces over the P2P network [14]. Hang et al [20,25] generate signature vectors based on low-level features describing peer content. Other techniques presented in [2, 19, 27] associate peers with semantic descriptions that can be simple keyword-based annotations, schema or ontologies. Mainly, clustering helps limiting the flooding, thus, the query is sent only to peers that are more likely to have relevant data objects.

In structured systems, DHT (Distributed Hashing Table) techniques have been proposed [10, 22, 26]. They aim to organize data in a key space to facilitate their access. Each peer in the network knows a given number of peers, and is assigned a unique identifier Id_{peer} using hashing key based on peer properties (e.g.IP address). Each published (shared) data on the P2P system is assigned a unique identifier Id_{data} using hashing key based on data content and/or name. Afterwards, the data is stored in and searched from the peer having the most simlar Id_{peer} to Id_{data}. DHT techniques allow an efficient data access with a complete lookup and reduce the hops number to locate data. However, DHT are more appropriate for exact queries. Thus, the main challenge for that systems is to process complex queries such as similarity, approximate and range selections. This challenge was recently addressed in [9, 21, 28] by adding a layer on top of the existing DHT systems to process complex queries.

Both clustering and DHT techniques have been intensively studied for different objectives. First, clustering organizes peers to optimize the search space and limit flooding and second DHT organizes data to maximize indexing efficiency for information retrieval. While DHT techniques assure a complete lookup but they are more appropriate for exact queries; clustering techniques support complex queries as similarity search but they generate approximate description of peers' content which may lead to a partial lookup.

3 Hybrid Overlay Network Architecture

Before introducing the density-based search algorithm, we present a hybrid architecture for clustering peers that share similar contents taking into account the semantic and low-level feature characteristics of peers content.

Figure 1 depicts the structure of the HON-P2P Hybrid Overlay Network, consisting of one or more overlays. The overlays are shown as boxes and clusters are shown as circles within the overlays. Each overlay includes one or more cluster of similar peers. We distinguished two types of overlay. The semantic overlays correspond to domain classification hierarchies which define concepts shared by the peer clusters. Similarly, feature-based overlays are created to represent low-level feature properties of data objects such as color, shape and texture. The architecture of an overlay cluster is a two level hierarchy consisting of super peers and simple peers. The super peers which are responsible for the management of the clusters have high processing and storage capacities.

As shown in figure 1, a peer can join more than one cluster in different overlays. To join a cluster, the peer must first carry out an overlay analysis to determine its semantic or feature-based representation. It then connects to any peer in the HON-P2P network to obtain overlay information. Once connected, the peer sends through the initial connection its representation to interested super peers which reply with their cluster representations. Finally, based on the returned cluster representations, the peer chooses to join and create links to one or more clusters. When a peer leaves the HON-P2P network, it notifies its directly connected neighbors and the super peers of its cluster groups.

We describe in details the feature-based overlays in HON P2P since the density-based algorithm proposed in section4 will be applied to them. However, more details description of semantic and feature-based clustering techniques used in HON-P2P network can be found in [12].

In the feature-based overlays, the HON-P2P system architecture is composed of three layers (figure 2). The data

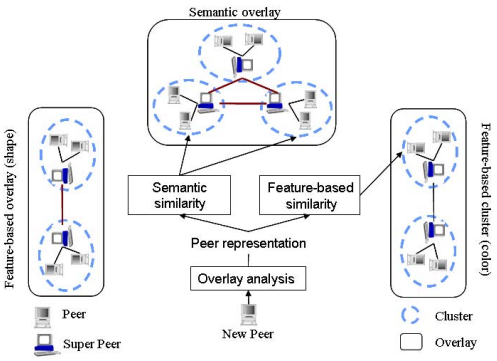

Figure 1. Hybrid overlay P2P network

organizing layer, the peer organizing layer and the clustering layer. The goal of HON-P2P is to group in the same clusters peers whose data are similarly distributed in a feature space defined by a set of low-level features. The functionalities of each layer of the system are based on the feature space. In the following we present in details the structure and the goal of each layer of the system starting first by defining the feature space.

3.1. The feature space

The feature space is defined by a set of low-level features $f_1, f_2, ..., f_k, ...f_n$ describing peers data. The idea is to define a partition of the feature space into cells and use the distribution of data over the cells as the basis for defining peer similarity, creating clusters and computing query similarities to peers and clusters. Thus, two peers are considered similar if their data are distributed on the same sub regions of the feature space.

To define a partition of the feature space into cells, we evenly divide the range of values $[f_i^{min}, f_i^{max}]$ of a feature f_i into m_i intervals of size $\lceil \frac{f_i^{max} - f_i^{min}}{m_i} \rceil$, for $i = 1, 2...n$. We denote the resulting set of cells $\phi = \{\varphi_1, \varphi_2, ..., \varphi_m\}$, where m is given by $m = \prod_{i=1}^{n} m_i$. For example figure 3 shows a feature space composed of two low-level features f_1 and f_2. Each feature is divided into 4 partitions. Thus, the number of the resulting cells is 4 * 4 = 16.

3.2 Data Organizing Layer

The data organizing layer represents the first step to build clusters in the HON-P2P system. It organizes peers data in the feature space. We consider a peer P_i and its data represented by a set of objects $O = \{O_{ij}\} = \{[f_{ij1}, f_{ij2}, ..., f_{ijk}, ...f_{ijn}]\}$, where f_{ijk} is the k_{th} feature value of O_{ij}. Each object O_{ij} corresponds to one point in the feature space, thus it is mapped to one cell. If we denote α_{ik} the number of objects of the peer P_i in the partition cell $\varphi_{k,k=1,2..,m}$ we can describe the content of P_i by a signature vector S_i defined over the feature space cells as $S_i = [\alpha_{i1}, \alpha_{i2}, ..., \alpha_{ik}, ...\alpha_{im}]$. The signature vector records the distribution of peer data objects over the cells. It defines the density of each cell with respect to the peer. This notion of cell density is used for mapping a peer to a cell and to create clusters. Figure 3 shows the partition cells of a 2-dimensional feature space using the low-level features f_1 and f_2 and the distribution of data objects of peers P_1, P_2, P_3 on cells.

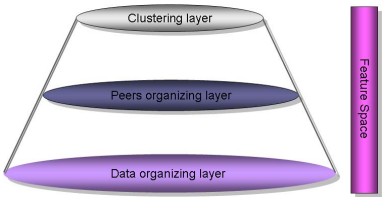

Figure 2. Architecture Layers

3.3 Peers Organizing Layer

The second layer consists in organizing peers in the feature space. Each peer is mapped to a set of cells containing its objects. The mapping is done using a threshold value T. A peer is mapped to a cell only if it has a number of objects higher than T in the cell. Once the peers are mapped in the feature space, we can compute the similarity between them. We propose a similarity of peer, called *Cell-similarity* defined as follows:

Definition 1 *(Cell-similarity)*
• *Two peers P_i and P_j are cell-similar with respect to a cell $\varphi \in \phi$ if they are both mapped to φ.*
• *This definition can easily be extended to define the cell-similarity of two peers P_i, P_j over a set $S \subset \phi$, of cells $S = \{\varphi_1, \varphi_2, ..., \varphi_r\}$. Two peers are cell-similar over a set S if they are cell-similar with respect to all the cells in S.*

In figure 3, peers P_2, P_3 exhibit cell-similarity on the cells $\varphi_{14}, \varphi_{15}$

3.4 Clustering Layer

The clustering layer creates clusters. To define a cluster, we consider a partitioning region of the feature space and group together all peers that are mapped to some cells of the region. A partitioning region consists of a set of cells in which each cell is adjacent to other cells of the region. Two cells are adjacent if they share a (d-1) dimensional hyperplane, where d represents the dimension of the feature space. The dimension d is equal to the number of low-level

Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)
0-7695-2585-7/06 $20.00 © 2006 IEEE

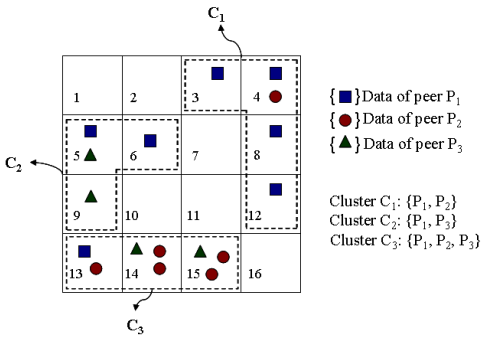

Figure 3. Partition cells and Clusters

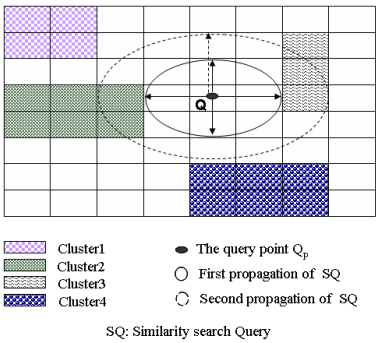

Figure 4. Example of similarity search

features describing the feature space. The adjacency between cells is used to create clusters in HON-P2P because it helps the similarity search based on range queries. Figure 3 depicts 3 clusters created using the density-based algorithm described in the next section. Peers P_1 and P_3 are in cluster C_3 but are not cell-similar even if both of them are cell-similar to $P2$. Both P_1 and P_3 can be used to process range queries submitted to cluster C_3.

4 Density-based Clustering and Similarity search

We propose a density-based clustering which is a variant of the Cluster-Forming (CF) algorithm presented in [15]. The focus of the density-based clustering are:

- Cluster adjacent cells for retrieving similar objects.
- Build clusters according to the cells density to provide a high recall.

4.1 Algorithm

To illustrate how the density-based algorithm works, we remind that the cell density represents the number of objects the cell contains. The density-algorithm works in the following way:
1- Compute the density of cells
2- Select the unmarked cells that have the highest density
3- A cell can be in one of three conditions:

- If there is no cluster adjacent to the cell, it forms a new cluster
- If there is one cluster adjacent to the cell, it joins the adjacent cluster
- If there is two adjacent clusters the cell joins the cluster having less density for load balancing. Note that the density of a cluster represents the number of objects it contains.

4- Mark the processed cells.
5- Repeat the process until all the cells are marked.

4.2 Similarity Search

Let a query object Q be described by the coordinates $[f_{q1}, f_{q2}, ...f_{qn}]$ in the feature space, where f_{qi} is the i_{th} feature value of Q. The query Q represents one point in the feature space called the query point Q_P (see figure 4). When a query Q is initiated, the requesting peer maps the query to the cell containing the query point Q_P where the required object resides called *target cell*. Then, the search takes the form of different actions, depending on if the target cell is found or not.

If the target cell is found, the requesting peer checks its cluster-index to extract the cluster address to which the target cell belongs. Then, the relevant super peer floods all the peers contained in the target cell to find the required object. When a peer receives the query Q, it computes the distance between the query Q and each of its objects belonging to the target cell. When the distance is less than a predefined *Similarity Threshold* ST, the object is sent to the requesting peer. The distance between a query and an object is defined as follows:

Definition 2 *(Query-Object Distance)*
Assuming that an object O is described by a set of features : D_o= $[f_{o1}, f_{o2}, ..., f_{ok}, ...f_{on}]$, and a query Q is described by a set of features D_q= $[f_{q1}, f_{q2}, ..., f_{qk}, ...f_{qn}]$. The distance between query Q and object O is given by $\sum_{i=1}^{n} \frac{|f_{qi} - f_{oi}|}{n}$.

If the target cell is not found, which means that the cell is empty, we then find the closest regions in the feature space to the target cell by generating a *Similarity search Query*

SQ. The similarity search query SQ is a range query having as lower and upper bounds l and u where the middle point (l+u)/2 is the query point Q_P. The similarity search query is propagated recursively to the adjacent cells until at least one cluster intersect the query SQ. The figure 4 shows an example of a query Q mapped to an empty cell. A similarity search query SQ then is generated and propagated to the adjacent cells. The query SQ in this example does not intersect any cluster because all the adjacent cells are empty. Therefore, the query SQ is extended with larger range values and propagated to the next adjacent cells. The second propagation of the query SQ intersects Cluster2 and Cluster3. Only the cells of Cluster2 and 3 covered by the query SQ are considered as *target cells*. These cells are then queried to return most similar objects to the initial query Q. Once the target cells are defined, the query SQ is then processed in the same manner described in the first case. Note that the similarity threshold ST for the query SQ is higher than the one of the initial query Q and it varies according to the user needs.

5 Evaluations

We have performed two parts of simulation to evaluate the HON-P2P system. In the first part we study the control parameters having an impact on clusters size and peers distribution in the feature space. The second part consists in evaluating the performance of similarity search using the density-based clustering.

5.1 Control parameters

The size and the number of clusters are controlled by six parameters:
• T: the threshold used to map peers to cells
• O: the number of objects
• N: the number of features
• P: the number of partitions used to divide each feature
• G: the cell granularity

We start by running a first simulation to show the threshold impact on clusters size. In this first part of simulation we run 10,000 peers with 1,500,000 objects following a uniform distribution where the average number of objects per peer is equal to 150. The feature space is described using 4 low-level features and composed of 10,000 cells.

Since the distribution of data is uniform, the peers objects are distributed in a random manner through cells, which reduces the probability for a peer to have a high density point in one cell. Therefore, if the threshold is higher than the maximum number of objects a peer P_i can have in one cell, P_i cannot be mapped into the feature space. In order to not loose the content of the peers which are not mapped into the feature space, we group them in a special cluster called *outlier cluster*. The figure 5 shows the impact of the threshold on the size of clusters and the outlier cluster.

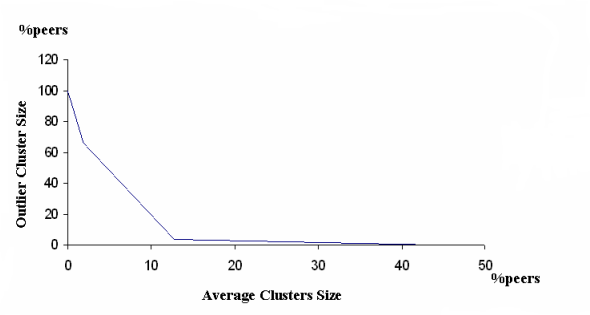

Figure 5. Threshold impact on clusters size

The average size of the clusters decreases when the size of the outlier cluster increases. An outlier cluster size equals to 0 means an efficient mapping of peers. We define a mapping efficiency measure M to evaluate the impact of the threshold on clusters size. Let $S_{outlier}$ be the size of the outlier cluster and NP be the total number of peers. The measure M is computed by ((NP-$S_{outlier}$)$\times$100)/NP.

The threshold value maximizing the mapping efficiency measure M depends on several parameters:
• *Objects number O*: the figure 6(a) shows the impact of the objects number on the threshold value allowing a complete mapping of peers into the feature space. We notice that with 500,000 objects, a threshold equals to 4 provides 7% of mapping efficiency, while with 2,000,000 objects the same threshold provides a high mapping efficiency equals to 80%.
• *Feature space dimension N*: The increase of the feature space dimension generates a larger number of cells, so the peers are mapped to cells with lower density which require a lower threshold as shown in figure 6(b).
• *Cell granularity G*: The cell granularity is defined by the number of partitions P. A low number of partitions generates a high cell granularity which increases the average number of objects by cell. Thus it tends to allow a high threshold for an efficient mapping as shown in figure 6(c).

5.2 Data distribution

Now we see how the threshold is affected by the data distribution. We consider two types of data distribution, the first one is a uniform distribution where each peer have equal chance to be mapped to any cell of the feature space. The second distribution follows a zipf law where peers are mapped to few cells of the feature space. The figure 7(a)

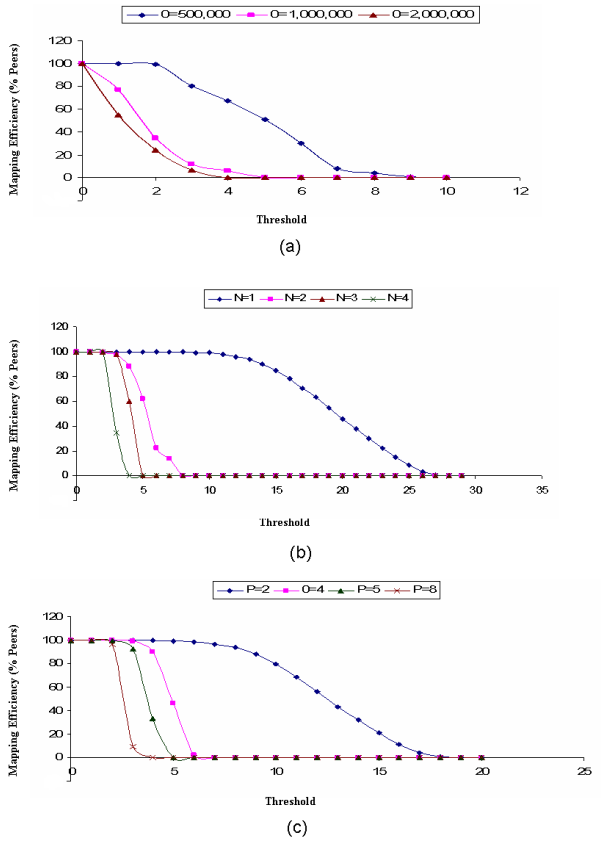

Figure 6. Control Parameters impact on the Threshold value

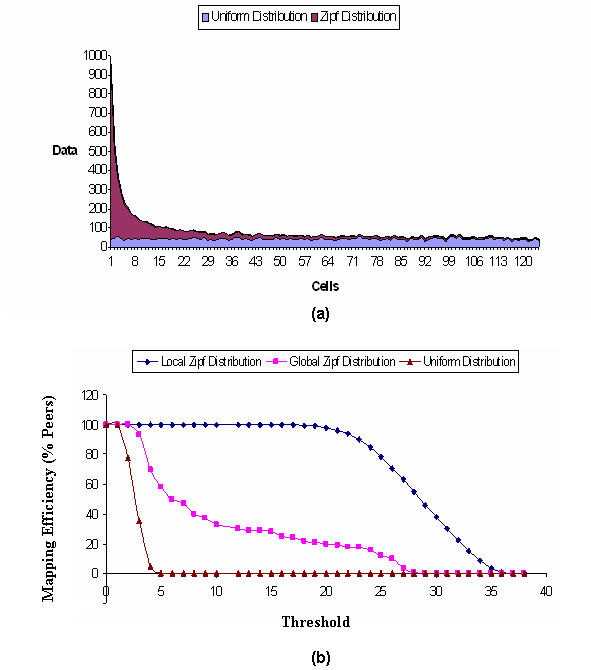

Figure 7. The impact of data distributions on the Threshold value

shows the distribution of data in the feature space. The cells have almost the same number of objects using a uniform distribution while the zipf distribution maps 80% of data objects to 20% of cells.

In our evaluation, we use the zipf distribution in two different ways:

• *Global Zipf Distribution*: All the peers of the system follow the same zipf distribution. Therefore, all peers will be mapped to a same region in the feature space with a high density point, where a region is composed of a set of cells.

• *Local Zipf Distribution*: Each peer follows its own zipf distribution. Each peer is mapped with a high density point to a region in the feature space and this region varies from a peer to another. Note that several peers can be mapped to the same region.

We run our simulations using a *uniform distribution*, a *global zipf distribution* and a *local zipf distribution* to analyse their impact on the threshold value providing an efficient mapping. Figure 7(b) shows that with a zipf distribution the system achieve high mapping efficiency even with a high threshold comparing to the uniform distribution. For example, a threshold equals to 6 generates 0,13% of mapping efficiency using a uniform distribution, while the same value provides an M value equals to 58% using a global zipf distribution. According to the results presented in figure 7(b), our HON-P2P system achieve a more efficient peers mapping with the local zipf distribution than with the global zipf distribution because it provides a high M value with high threshold.

5.3 Evaluation of similarity search

In this part of simulation we focus on three metrics to evaluate the similarity search:

• *The success rate*: the percentage of the most similar responses to the query object Q called relevant responses. The success rate is computed by R$\times$100/K, where R is the number of the relevant responses and K is the total number of responses.

• *The recall*: what fraction of the relevant responses has been retrieved? If a search only retrieves one hundred relevant responses out of three thousand that are available, that search has a low recall. If it retrieves all the available responses to the query Q, it has a high recall. Let TR be the total number of the relevant responses for a query Q and RR be the number of the retrieved responses. The recall is computed by RR$\times$100/TR.

Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)
0-7695-2585-7/06 $20.00 © 2006 IEEE

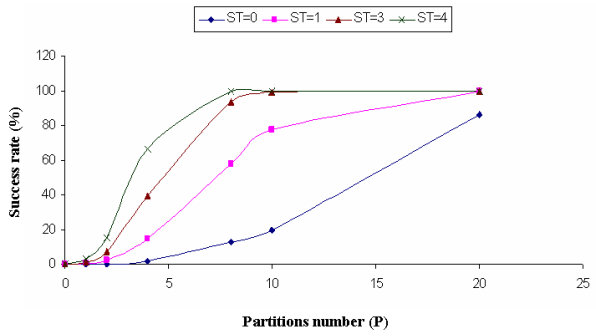

Figure 8. The impact of cell granurality on the success rate

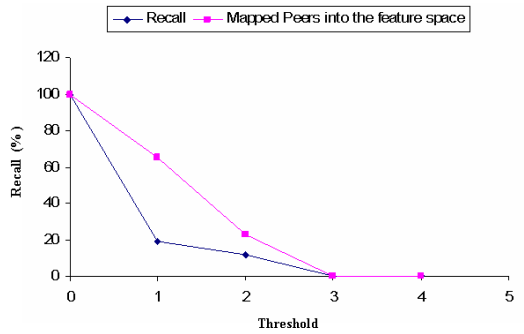

Figure 9. The recall according to threshold value

• *The query scope*: what fraction of peers in the system is involved in query processing? A smaller query scope increases system scalability. The query scope is computed by QP×100/P, where P is the total number of peers and QP is the average number of peers involved in the query processing.

We run a series of simulations with 10,000 peers, 1,500,000 objects following a uniform distribution, and 5,000,000 queries to measure the search similarity.

Figure 8 shows how the success rate depends on two parameters: the cell granularity and similarity threshold ST. The success rate reaches high values with low granularities compare to high granularities. In addition, when the similarity threshold increases, it provides a higher success rate. In our simulation, the range values of low level features have as lower and upper bounds f_{min} and f_{max}, where f_{min}=0 and f_{max}=40. As shown in figure 8 a partition equals to 1 and a similarity threshold equals to 4 provide only 2,96% of success rate, while a partition equals to 20 with similarity threshold equals to 0 allow 86,44% of success rate.

The recall in HON-P2P system depends on the used threshold value. If the threshold value T is equal to 0, peers are mapped to all the cells where their objects reside. In this case, a query Q will get the answers from all the peers in the system containing at least one object in the target cell. Therefore, the recall reaches 100% assuring a complete lookup. If the threshold value T increases as shown in figure 9, a peer having a number of objects in the target cell less than T will not be requested and its objects cannot be reached. Consequently, no complete lookup implies a recall decrease. In the figure 9, we notice that the recall decreases when the mapping efficiency value decreases.

The query scope depends on two parameters. First, the threshold value T. As presented before, the increase of the threshold leads to a low recall due to the fact that not all the peers having the relevant answers can be reached. Thus, the number of the peers involved in the query processing decreases with the increase of the threshold T. The figure10(b) shows the impact of mapping efficiency M on the query scope. Second, the cell granularity G. A high granularity increases the number of peers containing in each cell, thus the number of the requested peers increases as shown in figure 10(b).

6 Conclusion and future work

In this paper we studied the similarity search in the Hybrid Overlay Network HON-P2P. We proposed a similarity-search approach which organizes data and peers in a high dimensional feature space. We then introduced a density-based algorithm which groups the organized peers with similar data in the same clusters. Our simulations show the impact of the control parameters on clusters size and peers distribution. In addition, we evaluated the efficiency of the density-based algorithm to achieve a high success rate and recall. Our ongoing work in the HON-P2P focuses on: (1) Evaluating the performance of our approach taking into account the network parameters. Adding a physical layer will help to evaluate the average advantage ratio by computing the communication latency between two nodes belonging to the same or different clusters. (2) Evaluating the cost of building and updating the index tables when peers join and leave the network.

References

[1] A.Bestavros and S.Mehrotra. Dns-based internet client clustering and characterization. Technical Report BUCS-2001-

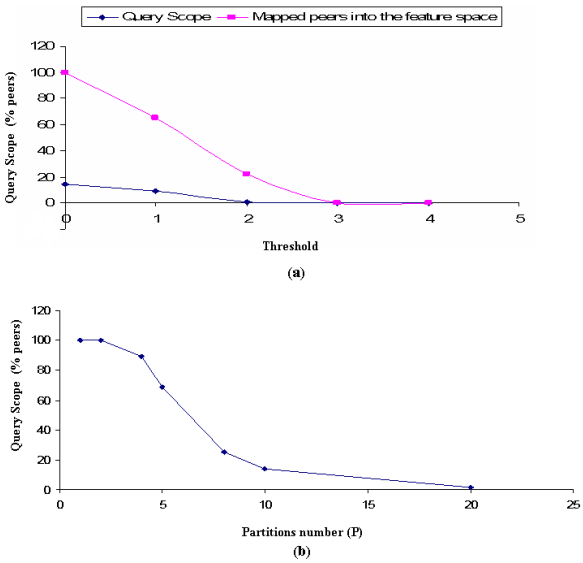

Figure 10. The query scope according the cell granurality and the threshold value

012 MA 02215, Boston University, Computer Science Department, Boston, June 2001.

[2] A.Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Computer Science Department, Stanford University, October 2002.

[3] A. Agrawal and H.Casanova. Clustering hosts in p2p and global computing platforms. *In Proceedings of the 3st International Symposium on Cluster Computing and the Grid (CCGRID)*, pages 367–373, 2003.

[4] Avaki. http://www.avaki.com/, 2001.

[5] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *In Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 177–190, 2002.

[6] eDonkey. http://www.edonkey2000.com, 2000.

[7] Gnutella. http://www.gnutella.com, 2003.

[8] S. B. Handurukande, A.-M. Kermarrec, F. L. Fessant, and L. Massoulie. Exploiting semantic clustering in the edonkey p2p network. *In Proceedings of the 11th ACM SIGOPS European Workshop*, 2004.

[9] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. L. S. Shenker, and I. Stoica. Querying the internet with pier. *Proceedings of 19th International Conference on Very Large Databases (VLDB)*, pages 321–332, 2003.

[10] I.Stoica, R.Morris, D.Karger, M. F.Kaashoek, and H.Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM*, pages 149–160, 2001.

[11] JXTA. http://www.jxta.org/, 2004.

[12] M. Kacimi, Y. Ma, R. Chbeir, and K. Yetongnon. Hon-p2p: A cluster-based hybrid overlay network for multimedia object management. *The 11th International Conference on Parallel and Distributed Systems, IEEE Computer Society*, 2005.

[13] KAZZA. http://www.kazaa.com/, 2002.

[14] K.Sripanidkulchai, B.Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems citation. *INFOCOM*, 2003.

[15] C. Li, E. Chang, H. Garcia-Molina, and G. Wiederhold. Clustering for approximate similarity search in high-dimensional spaces. *Transactions on Knowledge and Data Engineering*, 14(4):792–808, 2002.

[16] A. Loser, W. Nejdl, M. Wolpers, and W. Siberski. Information integration in schema-based peer-to-peer networks. *In Proceedings of the 15th Conference On Advanced Information Systems Engineering*, 2003.

[17] S. Marti, P. Ganesan, and H. Garcia-Molina. Sprout: P2p routing with social networks. *International Conference on Extending Database Technology (EDBT)*, pages 425–435, 2004.

[18] Napster. http://www.napster.com/, 2003.

[19] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Loser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. *Proceedings of the twelfth international conference on World Wide Web*, pages 536–543, 2003.

[20] C. Ng and K. Sia. Peer clustering and firework query model. *Proceedings of 11th World Wide Web Conference*, 2002.

[21] P. Rösch, K.-U. Sattler, C. Weth, and E. Buchmann. Best effort query processing in dht-based p2p systems. *Proc. of the 1st IEEE International Workshop on Networking Meets Databases (NetDB)*, 2005.

[22] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages pages 329–350, 2002.

[23] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. *In Proceedings of Multimedia Computing and Networking, San Jose, CA, USA,*, 2002.

[24] SETI@Home. http://setiathome.ssl.berkeley.edu/, 2001.

[25] K. C. Sia, C. Ng, C. Chan, S. Chan, and L. Ho. Bridging the p2p and www divide with discovir - distributed content-based visual information retrieval. *The Twelfth International World Wide Web Conference (WWW)*, 2003.

[26] S.Ratnasamy, P. Francis, M. Handley, R. Karp, and S.Shenker. A scalable content-addressable network. *In Proc. ACM SIGCOMM*, 2001.

[27] I. Tatarinov, Z.Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyska, G.Miklau, and P.Mork. The piazza peer data management project. *SIGMOD Record*, 3:47–52, 2003.

[28] P. Triantafillou and T. Pitoura. Towards a unifying framework for complex query processing over structured peer-to-peer data networks. *Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*, 2003.

[29] H. J. Wang, Y. Hu, C.Yuan, Z. Zhang, and Y.Wang. Friends troubleshooting network: Towards privacy-preserving, automatic troubleshooting. *IPTPS04*, 2004.