Cost Models and Performance Evaluation of Similarity Search in the HON P2P System

Mouna Kacimi and Kokou Yétongnon

University of Bourgogne, Laboratoire LE2I
Sciences et Techniques, 21078 Dijon Cedex France
{mouna.kacimi, kokou.yetongnon}@u-bourgogne.fr

Abstract. Similarity searching is particularly important in fully distributed networks such as P2P systems in which various routing schemes are used to submit queries to a group of relevant nodes. This paper focuses on the maintenance cost models and performances of similarity search in the HON P2P system, where data and peers are organized in a high dimensional feature space. We show through extensive simulations that HON has a low maintenance cost and is resilient to peers' failures.

1 Introduction

P2P systems and applications are gaining in popularity, spurred by the need for seamless interconnection of services and resources; distributing data indexes and processing among multiple nodes; and sharing large amount of data in dynamic ad hoc environments. Early popular file sharing P2P systems are based on simple exact keyword matching lookup and cannot meet the performance requirements of emerging applications. These applications often require complex range queries or content based similarity search on data such as images, text and video. The content of nodes is described by features, particularly physical image features that are represented in multidimensional data space.

Many search techniques have been proposed for P2P systems relying on their underlying overlay infrastructure. Flooding [2] is one of the first search techniques employed in P2P systems where each peer broadcasts the received query to directly connected peers. A Time-To-Live (TTL) mechanism or a random walk method can be used to reduce the number of peers that are involved in processing a query and avoid overloading the network. DHT systems [10] [9] [3] organizes data in a key space for efficient data access. Unique identifiers are assigned to the peers and the data. A data object is mapped to the peer with the closest identifier. Each peer maintains a routing table composed of its neighbors' identifiers. A lookup query routed to and processed by the peer that contains the corresponding data keys. DHT techniques are efficient for complete exact match queries but perform poorly for approximate similarity search. 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 [8] by adding a layer on top of the existing DHT systems to process multi-attribute range queries.

M. Guo et al. (Eds.): ISPA 2006, LNCS 4330, pp. 772–781, 2006.

In our previous work [6], we have presented a Hybrid Overlay Network (HON) for efficient similarity search. HON organizes both peers and data in an n-dimensional feature space based on content description. It is based on two key ideas. First it organizes and clusters peers sharing similar contents in the n-dimensional feature space to limit flooding overhead and send queries only to relevant peers. Second, it organizes and places similar data objects in relatively dense and adjacent regions of the feature space to achieve efficient processing of complex queries such as range and neighboring queries. The feature space represents particular attributes associated with data objects (e.g., color for an image, concept or keyword for text document) and is partitioned into cells obtained by dividing the range values of each feature into a number of intervals. Two data are similar if they are mapped to the same cell. The distribution of data objects over the cells defines the similarity between peers. Two peers are similar if their contents are distributed on the same sub regions of the feature space. We have presented extensive performance evaluation of similarity search quality in HON and shown that it achieves a high success rate.

The focus of this paper is on maintenance cost and fault tolerance issues of HON. Routing and localization methodologies are implemented in HON by maintaining partial routing tables in each peer, making the system very sensitive to membership changes. When peers join or leave the system, messages are exchanged to maintain the right P2P network topology. Thus, maintenance overheads and fault tolerance capabilities are important and can affect the performance of the system. The contributions of this work are two-fold: 1) we present and evaluate the HON system showing its scalability to large network size and numbers of data objects. Moreover, its adaptability to dynamic membership with low maintenance overheads. 2) we show through extensive simulations that HON efficiently routes queries along best available paths which make it resilient to peers' failures.

The remainder of the paper is organized as follows. In the next section, we give a brief description of HON. In section 3, we present the simulation setup describing the different parameters and metrics used to evaluate the system. Section 4 presents the evaluation results. Section 5 gives an overview about some related semantic-based search techniques to our approach. And finally section 6 concludes the paper.

2 HON

HON is a Hybrid Overlay Network that groups in the same clusters peers whose data objects are similarly distributed in a feature space defined by a set of features $f_1, f_2, ..., f_n$. The basic idea is to define a partition of the feature space into cells and use the distribution of data objects over the cells as the basis for defining peer similarity, creating clusters and computing query similarities to peers and clusters. Three steps are required to organize the data and peers in the feature space and create the clusters. First, the content of peers is distributed over the cells. Figure 1 shows the partition cells of a 2-dimensional feature space

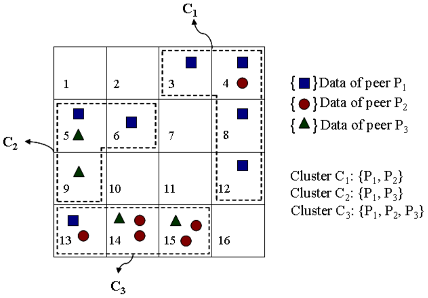


Fig. 1. Partition cells and Clusters

using the features f1 and f2 and the distribution of the contents of the peers P1, P2, P3 on the cells. Second, each peer is mapped into a set of cells containing its data objects. The mapping depends on a threshold value T. A peer is mapped to a cell only if it has a number of data objects higher than T in that cell. The last step is to create clusters by grouping the peers belonging to adjacent and dense cells in the feature space. We remind that two cells are adjacent if they share a (d-1) dimensional hyperplane, where d represents the dimension of the feature space. In addition, a cell density represents the number of objects the cell contains. The objectives of the clustering algorithm (called density-based) are to: 1) cluster peers belonging to adjacent cells for retrieving similar objects. 2) build clusters according to cells density to provide a high recall. Grouping peers within cells having high densities increases the number of retrieved similar objects. Clustering will not be discussed further in this paper. More details about the density-based algorithm and query processing can be found in our previous work [6]. The HON architecture contains two types of peers: *Super peers* and *Simple peers.* Super peers manage and maintain information about cells. Simple peers connect to super peers to process queries and have information about neighboring peers. Peers connection and disconnection to the network is processed in the following way.

Peer join: a new HON peer connects to the network and initiates the discovery of relevant super peers by broadcasting a *Join* descriptor containing its IP address and the set of cells to which its shared data belong. When a relevant super peer receives a *Join* descriptor, it updates the index tables and sends the new peer an *Accept-Join* descriptor with its IP address. Then, the new peer confirms its connection and builds the index tables with the information received from its super peers. Note that a new peer becomes the super peer of the empty cells to which it is mapped.

Peer leave: when a peer leaves the network, it sends a *Disconnect* descriptor to its super peers. A super peer receiving a *Disconnect* descriptor, removes all information related to the disconnected peer. When a super peer leaves the network, it first sends a *Disconnect* descriptor to its neighbors to initiate the choice for a replacement. Then, one of those neighbors takeover the cells of the disconnected super peer. The peers belonging to the corresponding cells update their index tables. In the case where the super peer disconnects suddenly from the network due to failure, there is a need for keeping in each peer information about the super peers of the neighboring cells.

3 Simulation Setup

We evaluate HON using extensive simulations that focus on the maintenance cost and failure tolerance. We use then several parameters and metrics. Parameters represent a set of measurable factors, such as *Threshold*, that determines the system behavior. Metrics are measurement functions that facilitate the quantification of some particular characteristics of the system such as the *Maintenance Cost*.

3.1 Parameters

Two sets of parameters are used in our simulation: *Control parameters* and *Workload parameters*.

Control: Parameters of control are used to build the system. Their values are specified before starting a simulation, and can change for each simulation to evaluate the system in different situations. Default values of parameters are defined if no particular specifications are given for the simulation.

Parameter	Description
T	represents the threshold value used to map peers to cells. It is initialized to 0 and varies between 0 and 50% of the average data object number per peer.
N	is the number of peers in the network. It varies from 2 to 2^{16}. Its default value is 2^{16}.
G	the cell granularity which is defined by the number of partitions of features. The number of partitions takes its values between 0 and 20 and the default value is 10.
O	the number of data objects per peer. It varies from 50 to 150 and its default value is 100.
D	is the type of the data distribution. It can be Uniform or Zipfian. The default value is the Zipfian distribution which reflects the real cases.

Workload: Parameters of workload are related to the occurring events in the system. They are specified when the system is build up to simulate a mixture of peer join, departure and search operations.

Parameter	Description
β	is used to simulate peers failure. It represents the percentage of failed peers in the system. β takes its values from 0% to 70% and its default value is initialized to 20%.
QN	is the average number of queries a peer sends over the network. Its default value is 500.
QD	specify the type of queries distribution. Similarly to data distribution, query distribution can be Uniform or Zipfian. Its default value is set to a Zipfian distribution.

3.2 Metrics

In addition of improving similarity search performance at a minimum overhead [6], we analyse in this paper other aspects of HON, such as maintenance cost and fault tolerance. Thus, we define the following metrics to evaluate system scalability and fault tolerance: *Maintenance Cost*, *Load Cost* and *Failure Cost*.

- *Maintenance Cost:* is the number of maintenance messages required to build the system. When peers join the network, we compute the number of exchanged messages between peers to update indexes. We consider M the total number of maintenance messages and N the number of peers in the network. The average maintenance cost is defined by M/N.
- *Load Cost:* represents the number of hops that maintenance messages require to reach the destination. Let H be the number of hops of all maintenance messages. The average load cost is computed by H/M, where M is the number of maintenance messages.
- *Failure Cost:* we simulate the failure of a specific percentage of peers after the network is build up. The percentage of failed peers varies between 10% and 70%. We then measure the ratio of searches that fail to find existing data objects in the network. Let TQ be the total number of queries and FQ the number of failed queries. The search failure ratio is computed by FQ/TQ.

4 Evaluation Results

We focus in our simulations on evaluating the maintenance cost and failure tolerance of HON. As the system size increases, the number of peers and cells might grow exponentially. Thus high number of indexes has to be maintained and updated when peers join and leave the network. Therefore, we need to carefully plan and consider the impact of system size increase and dynamic nature of peers on maintenance costs, thus scalability and user satisfaction. In addition, peers failures are common events in P2P systems. Hence a robust system needs to be resilient to theses failures.

4.1 Scalability

Overlay maintenance cost is proportional to the number of states maintained at each peer. We study the maintenance cost after a set of peers joining events. The maintenance cost depends on several parameters that are studied in this section to analyse the system behavior. We start running a first simulation where we vary the number of peers from 2 to 2^{16}. Each peer contains between 50 and 150 data objects following a uniform distribution. The feature space is described using 5 features and divided into 1024 cells. Then, we compute the average maintenance cost of simple peers, the average maintenance cost of super peers and last the average maintenance cost of total peers constituting the system.

Figure 2 shows that when the system starts with few peers, the average maintenance cost of super peers increases while the one of simple peers is null. This can be explained by the fact that when the system starts, all the cells are empty. Therefore, the joining peers are defined as super peers to manage those cells. As a result, almost no simple peers are present at the beginning of the system life. More peers join the network, more the number of empty cells decreases, thus the probability that a new peer will become a super peer decreases. Consequently, when no more peers are designed as super peers, the super peers cost will stabilize as shown in figure 2. Meanwhile, the average maintenance cost of simple peers start increasing and stabilize till new events occur in the system.

Figure 2 shows the average maintenance cost of total peers. We notice that it goes trough three steps: *Cost Increase*, *Cost Decrease*, and *Cost Stabilization*. The average maintenance cost increases when the number of super peers is increasing, decreases when simple peers join the network with their low cost, and stabilizes when the system is completely build up.

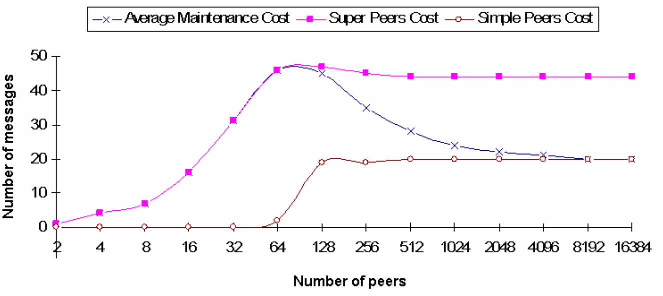
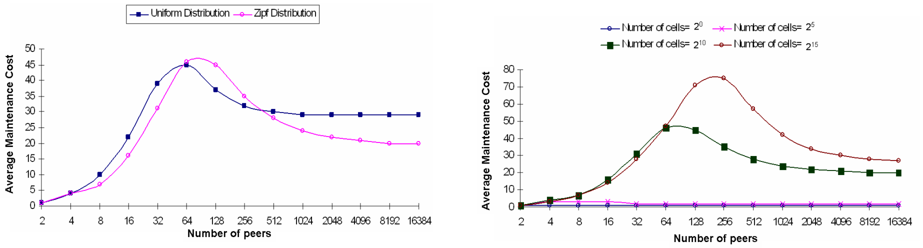


Fig. 2. The average maintenance costs

The average maintenance cost depends on several parameters. Here we study the impact of three parameters: *Data distribution*, *Cells Number* and *Threshold*. We start by the data distribution and we consider two types, the first one is a uniform distribution where each peer has 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 cells have almost the same number of objects using a uniform distribution while using a Zipf distribution 80% of each peer's content is mapped to 20% of cells.

We run our simulations using a *Uniform distribution* and a *Zipf distribution* to analyze their impact on the maintenance cost. As shown in figure 3-(a), The Zipf distribution assures lower average maintenance cost than the uniform distribution. Using a Zipf distribution a peer belongs to few cells in the feature space which means that it maintains few links to other peers in the system. While, with a uniform distribution, a peer may be mapped to a large number of cells that require a higher number of maintenance messages. For example, in figure 3-(a), the average maintenance cost when the system is build up is equal to 20 messages per peer using a Zipf distribution and 30 messages per peer using a uniform distribution. In the same way, we compute the maintenance cost according to the second parameter which is the number of cells. We notice that the number of cells depends on the dimensionality and the number of feature partitions. According to the results shown in figure 3-(b), the average maintenance cost increases with the number of cells.

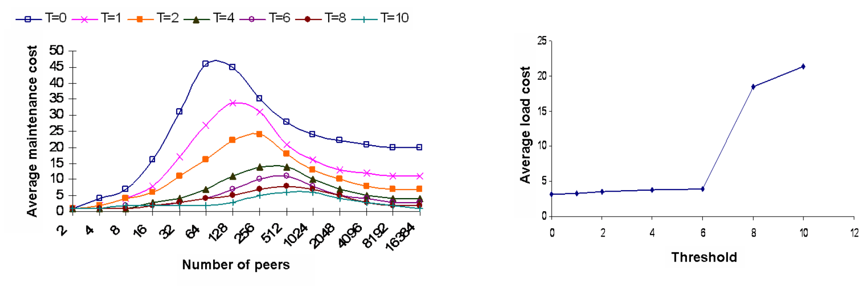


(a) Distribution impact on maintenance cost

(b) Number of cells impact on maintenance cost

Fig. 3. Distribution and number of cells impact on maintenance cost

The third parameter we use to evaluate the maintenance cost is the threshold value. We run our experiments using a Zipf distribution. Then we vary the threshold value used to map peers to cells and observe the system behavior. Figure 4-(a) shows that the average maintenance decrease when the threshold value increases. For example, when the threshold T increases from 0 to 10, the maintenance cost decreases from 20 to 2 messages per peer. It means when the threshold increases, it reduces the number of cells to which a peer can be mapped. Therefore, peers have fewer indexes to build and to update which reduce significantly the maintenance cost. On the other hand, when the threshold value increases, the load cost increases as shown in figure 4-(b). More hops are required when the peer gets less number of connections to other peers in the network by increasing the threshold value. The results presented in 4-(b) show that a threshold T=0 provides an average load cost equals to 3 hops per message, or a threshold T=10 provides a load cost equals to 21 hops per message.



(a) Threshold impact on maintenance cost

(b) Threshold impact on load cost

Fig. 4. Threshold impact on maintenance and load costs

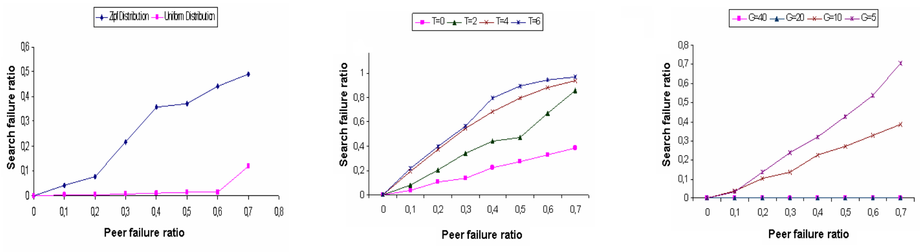
4.2 Tolerance to Failures

Each peer maintains a *state* parameter set to 1 or 0 to indicate respectively if the peer is online or offline. A failure simulation is given by setting the *state* of β peers to 0 and then start a set of search operations to compute the search failure ratio. We remind that β represents the percentage of failed peers and takes its values between 10% and 70%. We show in the following that HON is resilient to failures in spite of its hierarchical architecture. Since there are many different paths between two points in the feature space, when one or more of peer's neighbors fail, this peer can still route along the next best available path.

The fault tolerance of HON system depends on three parameters: *Data distribution*, *Threshold* and *Cells Granurality*. We consider first the data distribution parameter using uniform and Zipf distributions. According to figure 5-(a), we note that a uniform distribution improves the fault tolerance of HON system because a peer maintains a large number of links which increases the probability to reach the required destination when peers failures occur. For example, as shown in figure 5-(a), using a uniform distribution the search failure ratio reaches only 12% with 70% of failed peers, while it reaches 49% using a Zipf distribution.

The second parameter that has a great impact on the fault tolerance in HON is the threshold value. A high threshold reduces the number of links maintained per each peer. Thus, the increase of the threshold value implies an increase of the search failure ratio. As shown in figure 5-(b) using a Zipf distribution and a threshold T=0, 50% of failed peers results in a 37% of failed queries, while a T=6 provides a search failure ratio equals to 92%.

The last parameter that we studied to measure the fault tolerance is the cells granularity. Low granularities group a high number of peers in one cell which increase the probability to find data objects with lower hops number. Therefore, the search process has a low probability to fail. In figure 5-(c), we notice that a cell granularity equals to 40 assures a null search failure ratio and a cell granularity equals to 5 provides a search failure ratio of 71% with 70% of failed peers.



(a) (b) (c)

Fig. 5. Fault tolerance in HON

5 Related Work

Several deterministic semantic search approaches based on high dimensional space description of peers' content have been proposed. pSearch was the first system to allow decentralized, deterministic and non-flooding P2P information retrieval based on contents and semantics [1]. The main idea of pSearch is to store information of documents in DHT-based overlay network based on their representations. pSearch is based on CAN system[10] and uses latent Semantic Indexing LSI to generate semantic vectors for each document and query. These semantic vectors are used as index keys to store documents and route queries in the CAN space. pSearch aims to avoid the scalability problems of systems that are based on centralized indexing or index/query flooding. Even though each peer in pSearch maintains a large number of states (20), the search failure ratio grows rapidly with the number of node failures.

A similar approach to pSearch called MURK (Multi-Dimensional Rectangulation with Kd-trees) have been proposed by *Ganesan et al.* It uses a multi-dimensional data space that is partitioned into zones, where each zone is managed by one peer. A key difference between pSearch and MURK is that when a new pSearch joins a zone managed by an existing peer, that zone is divided equally between the two peers. Nevertheless, MURK splits zones into two parts of equal loads. In addition, the number of dimensions used by pSearch is governed by the dimensionality of the data while it is based on routing considerations in MURK. *Ganesan et al* focused on studying data locality properties and routing costs.

Li et al [7] have proposed a Semantic Small World (SSW) approach to facilitate efficient semantic based search in P2P networks. It is based on a semantic space where peers are clustered according to the semantics of their local data. These peer clusters are then self-organized into a small world network to assure an efficient search performance with low maintenance overhead. *Li et al* [7] have shown through extensive simulations that SSW is much more scalable to very large network sizes and very large numbers of data objects compared to pSearch. In addition, SSW assures good fault tolerance properties.

The multi-dimensional approaches presented above use mainly a maximum size M to define the boundaries of the feature space partitions. M represents the number of peers within the partition and is set to 1 in pSearch and MURK approaches. In HON, we use predefined partitions of the feature space to perform the similarity search giving more precise description of peers' content. The search is cell-based which assure an efficient accuracy using high granularities.

6 Conclusion

Similarity search plays a key role in information sharing in P2P systems. We have presented the main characteristics of the Hybrid Overlay Network (HON), a P2P system that organizes data and peers in a multidimensional feature space to allow efficient data search. We have evaluated HON using extensive simulations that focus on the maintenance cost and failure tolerance. We have shown the scalability of HON to large network size and numbers of data objects. Moreover, its efficiency to route queries along best available paths which make it resilient to peers' failures.

Our ongoing work focuses on studying the high dimensionality problems in HON, and the load balancing issues. Moreover, we are further tuning the performance of HON to measure the efficiency of the density-based algorithm in the dynamic environment of P2P networks.

References

1. C.Tang, Z. Xu, and M. Mahalingam. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In Proceedings of ACM SIGCOMM, pages 175-86, 2003.
2. Gnutella. http://www.gnutella.com, 2003.
3. I.Clarke, O.Sandberg, B.Wiley, and T.W.Hong. Freenet: A distributed anonymous information storage and retrieval system. In Lecture Notes in Computer Science, 2009:311-320, 2001.
4. 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.
5. JXTA. http://www.jxta.org/, 2004.
6. M. Kacimi and K. Yetongnon. Density-based clustering for similarity search in a p2p network. In proceedings of the 6th IEEE Symposium on Cluster Computing and the Grid (CCGrid'06), pages 57-64, 2006.
7. M. Li, W. Lee, and A. Sivasubramaniam. Semantic small world: An overlay network for peer-to-peer search. In proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04), pages 228-238, 2004.
8. P. Rosch, K.-U. Sattler, C. Weth, and E. Buchmann. Best effort query processing in dht-based p2p systems. Proceedings of the 1st IEEE International Workshop on Networking Meets Databases (NetDB), 2005.
9. 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 329-350, 2002.
10. S.Ratnasamy, P. Francis, M. Handley, R. Karp, and S.Shenker. A scalable content-addressable network. In Proceedings of ACM SIGCOMM, 2001.