Available online at www.sciencedirect.com

ScienceDirect

Information Processing and Management 44 (2008) 411–429

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Nearest neighbor search in metric spaces through Content-Addressable Networks ☆

Fabrizio Falchi [a], Claudio Gennaro [a,*], Pavel Zezula [b]

[a] ISTI-CNR, via G Moruzzu 1, 56124 Pisa, Italy
[b] Faculty of Informatics, Masaryk University, Brno, Czech Republic

Available online 1 May 2007

Abstract

Most of the Peer-to-Peer search techniques proposed in the recent years have focused on the single-key retrieval. However, similarity search in metric spaces represents an important paradigm for content-based retrieval in many applications. In this paper we introduce an extension of the well-known Content-Addressable Network paradigm to support storage and retrieval of more generic metric space objects. In particular we address the problem of executing the nearest neighbors queries, and propose three different algorithms of query execution. An extensive experimental study on real-life data sets explores the performance characteristics of the proposed algorithms by showing their advantages and disadvantages.

Keywords: Content-Addressable Network; Peer-to-Peer; Metric space; Similarity search; Nearest neighbor search

1. Introduction

Traditionally, search has been applied to structured (attribute-type) data yielding records that exactly match the query. A more modern type of search, similarity search, is used in content-based retrieval for queries involving complex data types such as images, videos, time series, text and DNA sequences. Similarity search is based on gradual rather than exact relevance using a distance metric that together with the database forms a mathematical *metric space*. The obvious advantage of similarity search is that the results can be ranked according to their estimated relevance. However, experience with the current, mostly centralized, similarity search structures reveals linear scalability with respect to the data search size, which is not acceptable for the expected dimension of the problem.

DOI of original articles: 10.1016/j.ipm.2006.04.002, 10.1016/j.ipm.2007.02.005.

☆ This work was partially supported VICE project (Virtual Communities for Education), funded by the Italian government, and by DELOS NoE, funded by the European Commission under FP6 (Sixth Framework Programme).

* Corresponding author.

E-mail addresses: fabrizio.falchi@isti.cnr.it (F. Falchi), claudio.gennaro@isti.cnr.it (C. Gennaro), zezula@fi.muni.cz (P. Zezula).

doi:10.1016/j.ipm.2007.03.002

Peer-to-Peer (P2P) architectures seem to solve the problem of scalability, and several scalable and distributed search structures have been proposed even for the most generic case of metric space searching (see Section 2 for a survey). They mostly concentrate on the *similarity range* queries where the execution algorithms satisfying (1) the autonomy of updates and (2) no central coordination or flooding strategies, are easier to implement. Since no bottleneck occurs, the structures are scalable and high performance is achieved through parallel query execution on individual peers (computer nodes).

Since the number of closest objects is typically easier to specify than the search range, users prefer *nearest neighbors* queries. For example, given an image, it is easier to ask for 10 most similar ones according to an image proximity criterion than to define the similarity threshold (i.e., the range), quantified as a real number. However, nearest neighbors algorithms are much more difficult to implement in P2P environments. The main reason is that traditional (optimum) approaches are based on a *priority queue* with a ranking criterion, which sequentially decides the order of accessed data buckets. In fact, an existence of centralized entities and sequential processing are completely in contradiction with decentralization and parallelism objectives of any P2P search network.

Capitalizing on our previous work of similarity range search through MCAN (Falchi, Gennaro, Zezula, & August, 2005), in this article we propose and experimentally test several nearest neighbor search algorithms. We first summarize the necessary background in Section 2, including the related work. Then in Section 3 we define the main properties of the MCAN. Section 4 describes alternative strategies for the nearest neighbor search, while the results of experimental testing are reported in Section 5. The paper concludes in Section 6.

2. Background

The most fundamental research results to our proposal are the Content-Addressable Network (CAN) (Ratnasamy, Francis, Handley, Karp, & Schenker, 2001a) as the storage infrastructure and the metric space concept as an abstraction of nearness (Chávez, Navarro, Baeza-Yates, & Marroquín, 2001). In the following, we provide the necessary background and survey relevant literature.

2.1. Content-Addressable Network (CAN)

The CAN is a distributed hash table that uses a function for mapping "keys" onto "values" defining positions of keys in the table. In the CAN, the table is represented by a finite set of peers. Each peer of the network is dynamically associated with a partition of an N-dimensional Cartesian space. Usually, the Cartesian space is an N-torus (i.e., the coordinate space wraps), and is targeted to store (X, V) pairs, where X is the "key" and V is the "value" associated with X. Assuming the "key" as a representation of content, basic operations of the CAN are insertion, lookup and deletion of respective (X, V) pairs. Formally, if we refer the domain of X as $\mathcal{D}$, we can define the mapping function G of the CAN as follows:

$$G : \mathcal{D} \rightarrow R^N, \tag{1}$$

where R^N is an hyper-rectangular region of $\mathbb{R}^N$.

The principle of the CAN is to divide the hyper-rectangular region R^N in a finite number of distinct rectangular zones, each of them associated with exactly one peer of the network. The peers are responsible for storing and searching of objects covered by their zones. Moreover, each peer is aware of the peers that cover adjacent zones, i.e., its neighbors.

Given a "key", the lookup function returns coordinates of the zone into which the key belongs. This is useful for insertion, deletion, and retrieval purposes. The search starts from an arbitrary peer of the CAN structure and proceeds by routing a message towards its destination by using a simple greedy forwarding to the neighbor with coordinates closest to the destination zone. In general, if we divide the R^N uniformly in h zones, each peer maintains 2N neighbors. Furthermore, the average routing path length is given by $(N/4)h^{(1/N)}$.

To simplify the discussion in the rest of the paper, we consider any element (key) of $\mathcal{D}$ as object, neglecting the fact that there is always a value V associated with it.