Master's thesis 2012-2013

Distributed Indexing of High-Dimensional Data

1 Context

An important research effort has been devoted during the last two decades to similarity search in high-dimensional data sets. Most of the approaches published so far consider centralized data structures. It is however widely acknowledged that scaling to very large data sets requires the distribution of both data storage and processing on a large number of coordinated servers. The workload can then be shared across the servers, and huge amounts of data can reside in the main memory of the participating machines, putting the design effort on the security, robustness and efficiency of network exchanges.

The internship will focus on the management of massive high-dimensional datasets in a distributed architecture. We specifically consider an infrastructure of data servers interconnected via a high-speed Local Area Network, organized in *clusters*. A cluster consists of a *master server* which handles coordination tasks, and many *servers* which share the storage and processing requests. A *Client* application connects to a cluster via a request to the Master, and obtains one or several addresses which can the be used to directly communicate with servers in the cluster. The architecture conforms to the general principles of *Scalable Distributed Data Structures* (SDDS) [3, 1] : (i) fair distribution, (ii) use of local caches in Client applications that stores an *image* (possibly outdated) of the whole structure, and (iii) dynamic expansion/shrinking. Let us briefly develop these principles in turn.

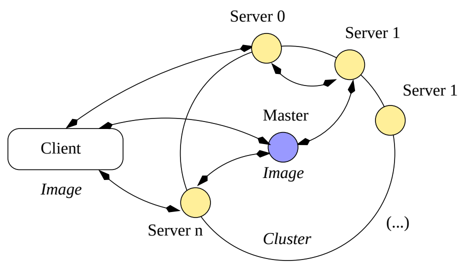

FIG. 1 – Network structure

No central directory is used for data addressing. This aims at achieving a good load balancing and fair participation of each server to the global storage and processing tasks. Using a centralized routing structure constitutes a bottleneck and must be avoided. Note that the Master server plays the role of a coordinator for some special and rare tasks (e.g., registering a new client or a new server). In most cases, Clients directly communicate with servers and simply ignore the Master which is therefore lightly loaded.

Distributed systems conforming to these principles are widely used. A well-known example is the Google File System which supports the Bigtable distributed range structure. Figure 1 summarizes the network structure. Since data is assumed to reside in main memory, efficiency is measured with respect to the number and the size of the messages exchanged between the participants (including the Master and the Client).

The characteristics of a SDD significantly differ from those of other distributed frameworks such as, for instance, P2P networks. In particular, servers can exchange data in order to achieve an efficient clustering and overall organization (whereas a P2P node comes with its own dataset), and we assume a full cooperation among the servers of a cluster via point-to-point or broadcast messages.

2 Objectives

Based on these characteristics we started to design a scalable distributed index that rely on the KD-tree principles (see [2]). Our index is a balanced binary tree with $2 * n - 1$ nodes mapped to a set of n servers. Each node N covers a subspace $N.coverage$ distinct from the subspace of *all* the other nodes (there is no overlapping). An internal node (or *routing node*) R defines a *split* of $R.coverage$ according to a dimension $i < d$ and to a position p. Finally each leaf node, or *data node*, stores a subset of the indexed objects along with a *routing table*. The set of leaf nodes coverages is a partition of the embedding space. *Routing tables* are used to forward queries that can not be answered locally to other data nodes. All queries will be forwarded only between data nodes with the help of of the routing tables. Information stored by an internal nodes is useless in this case and forwarding does not lead to any tree traversal.

The objective of the internship is twofold :
- First, after a thorough study of the state of the art, the candidate must study and validate our initial proposal. Based on this one, s/he must propose different algorithms for inserting/querying/deleting data. Different kind of searches could be considered : point query, window query, kNN-query, etc. An analytical model of these algorithms is expected ;
- Second, to validate this proposal, the candidate must implemented the index and to compare it with some relevant existing work.

Finally, we expect to publish this work in a high-level internal conference at the end of the internship.

3 Profile of the candidate

The candidate must have basic knowledges in the following areas : databases and indexing. Programming skills (in C/C++ or potentially Java) are compulsory.

4 Organization

The internship will take place into the ISID research team (http ://cedric.cnam.fr/isid) of the CEDRIC lab (http ://cedric.cnam.fr/). During this internship, the candidate(s) will collaborate also with Camelia Constantin, assistant professor at University Pierre and Marie Curie-Paris VI. This internship may be followed by a PhD.

5 Supervision

Cédric du Mouza, assistant professor, dumouza@cnam.fr, http ://cedric.cnam.fr/∼dumouza
Camelia Constantin, assistant professor, camelia.constantin@lip6.fr

Références

[1] C. du Mouza, W. Litwin, and P. Rigaux. SD-Rtre : A Scalable Distributed Rtree. In *Proc. Intl. Conf. on Data Engineering (ICDE)*, pages 296–305, 2007.

[2] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Softwares (TOMS)*, 3(3) :209–226, 1977.

[3] W. Litwin, M.-A. Neimat, and D. A. Schneider. LH* - A Scalable, Distributed Data Structure. *ACM Trans. on Database Systems (TODS)*, 21(4) :480–525, 1996.