# MULTIDIMENSIONAL VECTOR ROUTING IN A P2P NETWORK

Laurent Yeh, Georges Gardarin and Florin Dragan
*PRiSM Laboratory, Versailles University, 45 avenue des Etats Unis, Versailles, France*

Keywords: DHT, P2P, Routing, Multidimensional indexing.

Abstract: P2P systems tend to be largely accepted as a common support for deploying massively distributed data management applications. Many multidimensional data P2P indexing techniques suffer from severe limitations regarding the number of data dimensions to be indexed. In this paper, we propose a new approach for indexing multidimensional data in a P2P architecture. It is based on an efficient query overlay network (the so-called *routing layer*) built from a new data structure named *skip-zone*. We index data pieces of large dimensions as vectors and we adopt the polyhedral algebra for splitting the domain of possible values into sub-domains. To manage the overlay network meta-data required for tracking the network evolution or for routing data at each peer, we propose an efficient distributed *meta-data layer* that works in cooperation with the *routing layer*. An evaluation outlines the main properties of our architecture versus those of similar systems. The insensibility of the vector dimension in our data model is a key advantage of our approach.

## 1 INTRODUCTION

While first used for simple queries, such as searching for filenames (or document names), Peer-to-Peer (P2P) networks represent a possible basis for implementing large scale database systems. Among the main qualities that distinguish P2P networks, we remember: robustness, scalability, reliability, no central administration, no control over data placement. However, one of the problems of P2P data sharing systems is the weak data storage and retrieval capabilities when faced with anything more complex than simple values.

A recent problem that attracts the attention of the database community is how P2P networks can be used for efficient management of multidimensionnal data (e.g., vectors). Many vector-based applications require the indexing of long vectors. Long vectors can be used for representing image descriptors (for example, color descriptors for images are usually composed of at least 64 values). Yet another example is indexing XML documents with long paths. Each element name can be mapped to a value, the sequence of steps constituting a vector. The existing solutions (S. Abiteboul, 2005) (Cai et al., 2003) (Li et al., 2003)(Shu et al., 2005) have several limitations among we remember: due to the hierarchical organization of the zone encoding, wild-card queries require less or more flooding requests for each bit without determined values (0 or 1). The second problem is related to the key length of the DHT. For instance, Chord uses 160 bits; this is not enough for supporting long vectors. For a vector of dimension N and for a zone enclosing the point at level L, $N*L$ bits are required. Thus, the DHT entails a limit for the vector length.

The main contribution of this paper is a new P2P routing method suitable for vector data. The method is not sensitive to the dimension of vectors to be indexed (as shown in the section on experimental evaluation). The method is based on two basic ideas: the first is the use of polyhedral geometry (Laidlaw et al., 1986) that allows us to route independently of the considered dimension; the second is an extension of the Skip-List (Pugh, 1989)(Aspnes and Shah, 2003) data structure for fast routing. We name this extension *Skip-Zone*.

Our solution is based on a P2P architecture di-

Yeh L., Gardarin G. and Dragan F. (2007).
MULTIDIMENSIONAL VECTOR ROUTING IN A P2P NETWORK.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 486-489
DOI: 10.5220/0002382104860489
Copyright © SciTePress

vided in two layers: *i)* The first layer (named *metadata layer*) manages the metadata for the whole network. The main purpose of this layer is to compute the specific *zone* (a domain of values) controlled by each peer. *ii)* The second layer is a *routing layer* organized according to the new notion of *skip-zone*. These two layers cooperate for the network evolution: the first layer provides metadata information for every peer that connects and disconnects into the network, or splits its content; the second layer efficiently routes the requests to the relevant vectors. The main advantage of our solution is that peer evolution (splitting and merging contents) involves only local metadata. In contrast, other algorithms publish the splitting metadata to every peer in the overlay network for each peer splitting event. This appears as a bottleneck for the network scalability.

## 2 NETWORK STRUCTURE

We propose a network overlay divided in two layers. The first (metadata layer) deals with metadata; it insures the splitting of the value domain in a sequence of zones covering a continuous range of values. The second layer deals with query routing.

### 2.1 Network Architecture

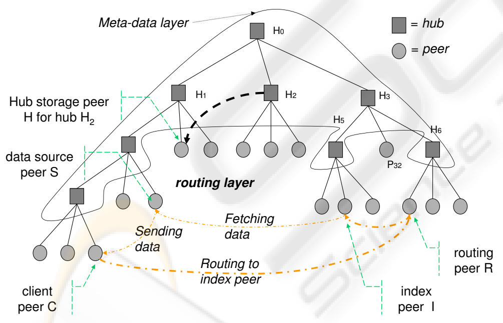

Figure 1: A Network Instance.

Figure 1 presents an instance of the P2P network where the metadata layer is portrayed in the upper part and the routing layer in the lower one. The metadata layer is arranged in a tree-like topology and manages how data is distributed to peers. The routing layer manages the routing of data requests to peers with relevant data. It integrates the data source peers as tree leaves. Peers are here data storage elements. In this organization, each peer is placed according to its index role (e.g., peer I). A peer is located in the network under the index node determined by the domain of values it manages. The localization of a peer index is done by the metadata layer as explained in the sequel.

### 2.2 Metadata Layer

The Metadata layer deals with the problem of splitting the domain of possible values into *sub-zones*. Hubs are organized in a hierarchy and each leave zone is allocated to an index peer.

Thus, zones are divided along the descending paths in the hub hierarchy; the zones from upper levels cover zones from lower levels. For example according to Figure 1 the zone indexed at the hub $H_0$ describes all the domain of values. The union of zones described by the hubs $H_1, H_2, H_3$ represents the zone described by the root hub $H_0$. An other example is $Zone(H_3) = (Zone(H_5) \cup Zone(P_{32}) \cup Zone(H_6))$. Recursively, each hub describes a sub-zone of its parent one. By using a tree topology, zones are split continuously. Each index peer can store any value described by its immediate parent hub.

**Structure of a hub:** A hub is a metadata-layer component that helps to define the domains of values for every peers. Each hub has $N$ children (that are other hubs or peers) and one parent hub, except the root hub. A hub of size $N$ has a single input and N-outputs and is characterized by a set of zones $\{Z_1, Z_2, .., Z_n\}$.

Each zone is defined by a polyhedron (Laidlaw et al., 1986); a polyhedron is able to describe a "volume" in any dimension; it provides the $\in$ operator for testing if a point is included in a volume. Mathematically, a polyhedron is defined by a set of inequations. An example in dimension 4 is: $x_1 \geq 10 \wedge x_1 \leq 20 \wedge x_2 \geq 8 \wedge x_2 \leq 167 \wedge 2*x_3 + x_1 \geq 20 \wedge x_3 \leq 120 \wedge x_4 + 5*x_2 \geq 10 \wedge x_4 \leq 20$. By hub-dimension, we refer to the maximum number of children that can be connected to a hub. The hub-dimension is a parameter that is considered to be fixed for all hubs in the metadata layer (a strategy with a variable hub-dimension can also be considered).

Two important properties of a set of zones stored in hubs are:
*(i)* $Z_1(HUB_k)$ is_adjacent $Z_2(HUB_k)$ is_adjacent ... is_adjacent $Z_n(HUB_k)$.
*(ii)* $\exists j$ such that $(Z_1(HUB_k) \cup Z_2(HUB_k) \ldots \cup Z_n(HUB_k)) \subset Z_j(HUB_f)$ where $Parent(HUB_k) = HUB_f$.
The first property (adjacency property) assures the continuity of zones, while the second states that each set of zones described by a parent hub is included in a zone described by a hub at a higher level.
**Dynamicity of the Metadata Layer:** The metadata layer evolves in terms of hub splitting and grouping.

A hub is split when its load is too high. Two twin hubs are tentatively grouped when the load of one is below a threshold. A peer can trigger either a split or a group without centralized decision. Depending on the data load of their peer children, hubs may split or may fusion.

**Peer index split**: When a peer $P$ is saturated, its load must be distributed to other peers. For that P invites an *available peer* from the network to share a part of its load, then the peer P shares its load with the new index peer. If the *hub-dimension* of the hub $H_i$ allows to receive a new zone, this new peer is simply linked to the old hub. Otherwise, a new hub is created and the two previous peers are connected to it.

**Computation of New Zones:** Splitting a peer load with load balancing control is an important problem. The problem is to find a set of inequations that divides the set of $n$ multidimensional points indexed by a peer in two sets of approximately equal size. Splitting in two even subsets is not an easy task. One simple solution is to iterate through the set of vectors for each dimension until a set of good splitting equations is found. Two polyhedra containing approximately the same number of points must be created from the original one. As an example, consider a network indexing vectors of 3 dimensions $< x1, x2, x3 >$ and a peer $Pi$ indexing the set of 4 points: $\{v_1, v_2, v_3, v_4\}$=$\{< 0,0,0 >, < 0,0,2 >, < 0,2,0 >, < 2,0,0 >\}$. Considering the values of the first dimension, a possible split is defined by $x_1 = 2$. In this case, the decomposition generates a set of 3 values and a set of one value. As it can be observed the values are not evenly distributed and another decomposition must be found. Similarly, we can find decompositions on the second and on the third dimensions with the same (uneven) distribution of values. To find a good solution, at least two dimensions must be considered.

## 2.3 Routing Layer

This section describes the routing layer which is run for routing a data request through the network. It does not require the metadata layer. Every peer encloses a data structure named *skip-zone* that can work autonomously.

### 2.3.1 Skip-Zone Structure for Routing

Figure 2 illustrates the architecture of the vector routing layer. It is inspired from traditional skip-list. The routing layer is built over all peers in the lower part of the network of hubs as illustrated in figure 1.

Every peer has an autonomous Skip-Zone. A Skip-Zone uses the notion of zone again as illustrated in Figure 2. Every peer has $2*L$ zones ($L$ zones for the left part and $L$ zone for the right part). $L$ is the number of levels in a skip-zone structure, which is a fixed parameter for the entire network. According to the Figure 2, the skip-zone defines three levels. On each side, a zone covers a set of peers. Notice that every peer is localized in the network according to the position given by the metadata layer. For a peer, a zone defines a domain (space) of multidimensional values indexed by all peers from its position to a peer at a distance $D^k$, where $k \in 1..L$.

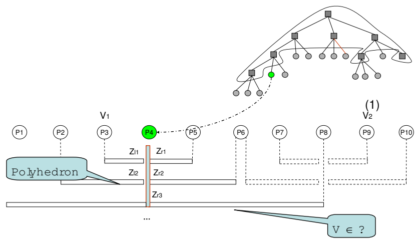

Figure 2: The Routing Layer.

Thus, the first level covers all peers at a distance $D^1$ from the original peer (e.g. $Zl_1$, respectively $Zr_1$). The second level covers a distance of $D^2$, the third level for a distance $D^3$, and so on. The choice of $D$ is arbitrary, but it is fixed for the entire network. According to Figure 2, and based on the ordering relation, the $Zr_1$, (respectively $Zl_1$) zone is included in the zone represented by $Zr_2$ (respectively $Zl_2$). We have the following relationship: $Zl_i \subseteq Zl_j, Zr_i \subseteq Zr_j$ for $i < j$.

Each zone is composed of a polyhedron and a pointer to the peer at the extremity of the zone. For examples, the zone $Zr_2$ has a pointer to the peer P6, and the zone $Zr_3$ has a pointer to the peer P10.

The number of levels for a skip-zone could be high. Indeed, the storage cost for each polyhedron is very low (2 inequations for each dimension). The computation of each zone is inexpensive as described in the above section. Thus, for a network, it makes sense that a Skip-Zone can cover the entire network or a large part of it.

## 3 MULTIDIMENSIONAL POINT ROUTING

Before presenting the routing principles, notice that according to our previous description of the metadata layer, if a point exist, it must be in one and only one index peer (or in other words, in a single zone).

Considering the example in Figure 2, for a point V to be routed to the relevant peer index, the algorithm must firstly checks whether it is in the zone $Zr_3$. If it is true, the immediate superior level ($Zr_2$) is used to determine if the point V is at the left or right of P6. For example, in the case that the point is at the right of P6, the request is propagated to P6 which recursively processes the algorithm. Notice that in a given skip-zone, the search cost is logarithmic.

On the contrary, if the answer of the previous check in $Zr_3$ is false, the point is tested again with the left highest level $Zl_3$ with the process described in the previous paragraph. In the case that the check is again false, the request is forwarded to the extremity of $Zl_3$ and $Zr_3$ (peer $P_8$). The network clones the original request in two requests but with a notable difference: Each request has a direction (go left or right). In our example, only P8 will receive the propagation of the request. At $P_8$, due to the direction, the check is only done on the right part of the skip-zone owned by P8. If the test is again false, the request is once more propagated to the skip-zone at the highest right extremity.

## 4 DIMENSION INSENSIBILITY

In contrast to other works, in which the DHT routing key size represents a bottleneck, our routing approach is not very sensible to the dimension of the searched vector. Indeed, our routing principle is based on the ($\in$) operation that checks if a point is in a zone (volume in a N-dimension space) or if two volumes intersect. This check is processed according to the polyhedral principle that requires 2*D tests. Each check is the test of a set of inequations of the type $a_d x^d + a_{d-1} x^{d-1} ... < 0$ which is solved with a very low CPU cost compared to messages transmission delay or disk access. Thus, each new dimension requires to test at least two additional inequations. Solving for data of 1 dimension or N dimensions differs only by the CPU time required to check each inequation. That is why our approach is almost not sensible to the dimension of vectors.

## 5 CONCLUSION

In this paper, we introduced a P2P system suitable for indexing and routing vector data. Compared to other works, our originality is twofold. First we use polyedra for describing data zone managed by each peer. This is done in a first overlay network named metadata layer. Secondly, we extend the principle of skip-list for working with data of any dimension. We named this new structure ”skip-zone”.

## REFERENCES

Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M., and Schmidt, R. (2003). P-grid: a self-organizing structured p2p system. *SIGMOD Record*, 32(3):29–33.

Aspnes, J. and Shah, G. (2003). Skip graphs. In *Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 384–393.

Bharambe, A. R., Agrawal, M., and Seshan, S. (2004). Mercury: supporting scalable multi-attribute range queries. In *SIGCOMM ’04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 353–366, New York, NY, USA. ACM Press.

Cai, M., Frank, M., Chen, J., and Szekely, P. (2003). *MAAN: A Multi-Attribute Addressable Network for Grid Information services*, volume 00. IEEE Computer Society, Los Alamitos, CA, USA.

Laidlaw, D. H., Trumbore, W. B., and Hughes, J. F. (1986). Constructive solid geometry for polyhedral objects. In *SIGGRAPH*, pages 161–170.

Lawder, J. K. and King, P. J. H. (2000). Using space-filling curves for multi-dimensional indexing. *Lecture Notes in Computer Science*, 1832:20–??

Li, X., Kim, Y. J., Govindan, R., and Hong, W. (2003). Multi-dimensional range queries in sensor networks. In *SenSys ’03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 63–75, New York, NY, USA. ACM Press.

Morris, R., Karger, D., Kaashoek, F., and Balakrishnan, H. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM 2001*, San Diego, CA.

Pugh, W. (1989). Skip lists: A probabilistic alternative to balanced trees. In *Workshop on Algorithms and Data Structures*, pages 437–449.

S. Abiteboul, I. Menolescu, N. P. (2005). Peer-to-peer warehousing of xml resources. ICDE.

Shu, Y., Ooi, B. C., Tan, K.-L., and Zhou, A. (2005). Supporting multi-dimensional range queries in peer-to-peer systems. In *P2P ’05: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P’05)*, pages 173–180, Washington, DC, USA. IEEE Computer Society.