GROUPINGS OF NODES FOR DISTRIBUTED DATABASES

Mădălina Văleanu and Grigor Moldovan

Abstract. The grouping and the reorganization of distributed databases in many applications can be a significant cost issue. In this paper we define a function d_{ij} that verifies the distance properties and that can contribute to optimize the databases grouping in a computer network.

1. Introduction

Distributed databases are associated to computer networks possessing a certain number of nodes. In its turn, a computer network overlaps a non-oriented graph $G = (N, U)$, where $N = \{N_1, N_2, \ldots, N_n\}$ is the set of nodes, and $U = \{[a, b]/a, b \in N\}$ is the set of edges.
Be $B = \{B_1, \ldots, B_m\}$ a distributed database in the nodes of the computer network. In one of the network nodes, there can be found one, more or none of the databases $B_i, B_i \in B, i = \{1, \ldots, m\}$.
An information system that manages data from the database B_i , called Distributed Information System, noted DIS, will have performance if it will take into account the architecture of the computer network which obviously depends on the particular aspects of the graph $G = (N, U)$. One modality to optimise data processing through repeated accessing of the data can consider grouping, i.e. their zoning following various criteria. In previously published papers [MOL92,93], zoning related to various criteria, such as equity, compactness, contiguity and enclave exclusion was discussed and studied. Analysis of some procedures (algorithms) through they can be achieved was performed.
The optimisation of data processing can also be performed by modifying the placement of databases in the nodes of the computer networks so that the response time to frequent requirements in a SID is the smallest possible [

MV06,08]. The two previously mentioned approaches are deterministic in nature. Practically, the processing of data in a distributed database is generally random in nature. This occurs in almost all distributed databases management systems. Hence, a statistical approach, respectively a probabilistic approach, is required. These approaches will be investigated as follows.

2. Some notion

Let us define some notions of use in the following.
Definition. *Be the distributed database $B = \{B_1, B_2, \ldots, B_m\}$ and the set of nodes $N = \{N_1, N_2, \ldots, N_n\}$ belonging to a computer network. The cluster of a database $B_i, B_i \in B$ is $K_i, K_i \subseteq N$ whose elements are nodes from which this database is accessed.*
Observations.

a) There is a biunivocal relationship between the databases B_i of a distributed database B and their clusters, i.e. $B_i \leftrightarrow K_i, i = \{1, \ldots, m\}$.

b The set of all the clusters relative to the distributed database is

$$K = \{K_j / K_j \subseteq N, j = \overline{1, 2^m}\}$$

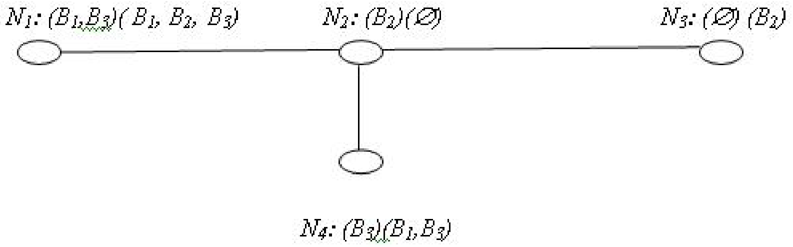
c) cluster is served by a server Sk obviously placed in a node N_s.

Let us consider a Distributed Information System (DIS) concerned with the management of a distributed database $B = \{B_1, B_2, \ldots, B_m\}$ and the databases found in the nodes $N = \{N_1, N_2, \ldots, N_n\}$ of a computer network. In every node $N_s, N_s \in N$, queries are formulated and they suppose theat some databases are accessed from B. With respect to the location of these databases, two situations can be distinguished:
- databases found in the respective node, that is they have the same location; they will be specified by notation $(B_{i1}, B_{i2}, \ldots, B_{ip})$; - databases accessed from the respective node, having other locations that the node in question: these are specified with the notation $(B_{j1}, B_{j2}, \ldots, B_{jq})$;
Consequently, it results from the aspects mentioned above, that for node Ns one can use the following notation: $N_s : (B_{i1}, B_{i2}, \ldots, B_{ip})(B_{j1}, B_{j2}, \ldots, B_{jq})$. In the Figure below, an example of a four node network $N = \{N_1, N_2, N_3, N_4\}$ is given , the distributed database $B = \{B_1, B_2, B_3\}$ and the structure of the queries in every node, specified near every node.

From the figure, one can see that:

$$K_1 = \{N_1, N_4\}, K_2 = \{N_1, N_3\}, K_3 = \{N_1, N_4\}.$$

It also results that $K_1 \equiv K_3$, hence, a merging o the bases B_1 and B_3 can be performed. When designing a DIS that uses databases, our main objective is


N_1: $(B_1,B_3)(B_1, B_2, B_3)$
N_2: $(B_2)(\varnothing)$
N_3: $(\varnothing)$ (B_2)
N_4: $(B_3)(B_1,B_3)$

to efficiently exploit the databases found in the nodes of a computer network in so far queries occurring as time goes on are concerned. In this sense, the grouping of databases in certain nodes could be useful. The grouping should be made so that those participating with the highest probability together with the response to be given to some queries to situated as nearly as possible versus one another.

Due to the requirements mentioned earlier, one finds that the approach of this issue should use stochastic, not exclusively deterministic, principles as mentioned above.

3. The extent of databases access degree

During the exploitation of a DIS, certain queries are repeated put and used. These queries suppose the use of distributed databases. After a certain time, statistically, one can determine the relative frequency with which Bk is used to perform every query. In a SID, let $C = \{C_1, C_2, \ldots, C_s\}$ be the set of queries and $f_k, k = \overline{1, s}$ the frequencies with which the databases $B_k, B_k \in B; k \in \{1, 2, \ldots, m\}$ are used. If the statistical sample asks for a number of n queries, and the database B_k was used at an absolute frequency of nk times, then the relative frequency will be $f_k = \frac{n_k}{n}$. In the case of these relative frequencies, there is the property: $\sum\limits_{k=1}^{s} f_k = 1$.

The query C_k is triggered by the node Ns and uses a subset of the set of databases from the distributed database, that is $(B_{i1}, B_{i2}, \ldots, B_{ip})$ as well as other databases in other nodes $(B_{j1}, B_{j2}, \ldots, B_{jq})$. To every database B_i from the distributed database B we will associate, within a query C_k,, a number that represents a piece of information regarding the use of the respective database

B_i in the query C_k, information noted $m(C_k, B_i)$; by definition this number is:

$$m(C_k, B_i) = b_i \cdot f_k \ k = \overline{1, s}, i = \overline{1, m}$$

where b_i represents the number of times B_i was accessed.
Observations.

a) If $m(C_k, B_i) = 0, k = \overline{1, s}$, that no access is required from B_i then a situation of degeneration is met for B_i and we say that Bi is in a state of degeneration.

b) If $m(C_k, B_i) = m(C_k, B_j), k = \overline{1, s}$ then B_i and B_j are similar.

c) A degenerated database is similar to any other degenerated database.

Definition.*Be $B_i, B_j \in B$. We define the values:*

$$q_{ij} = \sum_{k=1}^{s} \min\{m(C_k, B_i), m(C_k, B_j)\}; i, j = \overline{1, m}.$$

Observations.

a) If $i = j$ then $q_{ii} = \sum_{k=1}^{s} m(C_k, B_i) = \sum_{k=1}^{s} b_i f_k = b_i \sum_{k=1}^{s} f_k = b_i, i = \overline{1, m}$ represents the total number of accessing of the database B_i.

b) We have $\forall i, j = \overline{, m}, q_{ij} \geq 0$. This property results immediately from the fact that $m(C_k, B_i) \geq 0; \forall i = \overline{1, m}, k = \overline{1, s}$.

c) We have $\forall i, j = \overline{q, m}, q_{ij} = q_{ji}$. This property results immediately from the definition of the values.

d) We have $\forall i, j = \overline{1, m}, q_{ij} = max_j q_{ij}$. Actually

$$\begin{aligned} q_{i,j} &= \sum_{k=1}^{s} \min\{m(C_k, B_i), m(C_k, B_j)\} \\ &\leq \sum_{k=1}^{s} m(C_k, B_i) = \sum_{k=1}^{s} \min\{m(C_k, B_i), m(C_k, B_i)\} = q_{ii}. \end{aligned}$$

Hence $q_{ii} = \max_j q_{ij}$

e) Function $q(i,j) = q_{ij}$ is not transitive. This statement can be verified easily in the example given previously in a particular case.

Definition. *Be* $\forall i,j = \overline{1,m}; p_{ij} = \frac{q_{ij}}{q_{jj}}$. *These magnitudes represent the probabilities of accessing database* B_i *in a query addressed to database* B_i.
Now we shall define the distance between two databases B_i, respectively B_j accessed during a query C_k.
Definition. *Be* $C = \{C_1, C_2, \ldots, C_s\}$ *and* $C_k, C_k \in C; k \in \{1,2,\ldots,s\}$. *Be then,* $B = \{B_1, B_2, \ldots, B_m\}$, *and* $B_i, B_j; i,j \in \{1,2,\ldots,m\}$ *two elements of this set. In relation to databases* B_i *and* B_j *and with respect to query* C_k *we define distance:*

$$d_{ij} = \sum_{k=1}^{s} \|m(C_k, B_i) - m(C_k, B_j)\|; i,j \in \{1,2,\ldots,m\}$$

Now we shall establish some important properties of distance d_{ij}.
Theorem. *Distance* $d_{ij}; i,j \in \{1,2,\ldots,m\}$ *has the following properties:*

a) $\forall i,j \in \overline{q,m}; d_{ij} \geq 0$;

b) When in B similar databases lack, then $d_{ij} = 0$ *occurs, only and only if* $i = j$;

c) $\forall i,j \in \overline{q,m}; d_{ij} = d_{ji}$;

d) $\forall i,j,p = \overline{q,m}; d_{ij} \leq d_{ip} + d_{pi}$ *(triangle inequality).*

Proof:

a) As $\forall k = \overline{1,s}, \forall i,j \in \overline{1,m}; \|m(C_k,B_i) - m(C_k,B_j)\| \geq 0$, it yields from the definition of the distances considered, that we have $d_{ij} \geq 0$.

b) Be $d_{ij} = 0$, i.e. $\sum_{k=1}^{s} \|m(C_k,B_i) - m(C_k,B_j)\| = 0$. This equality takes place if and only if $\|m(C_k,B_i) - m(C_k,B_j)\| = 0$, i.e. $m(C_k,B_i) = m(C_k,B_j)$ and B_i, B_j are similar. With an inverse reasoning, we come to the conclusion that the property stated is true.

c) The inequality of the triangle in full form is written

$$\sum_{k=1}^{s} |m(C_k, B_i) - m(C_k, B_i)| \leq \sum_{k=1}^{s} |m(C_k, B_i) - m(C_k, B_p)| + \sum_{k=1}^{s} |m(C_k, B_p) - m(C_k, B_j)|$$

This inequality occurs if:$\forall k = \overline{1, s}; \forall i, j, p = \overline{1, m}$

$$|m(C_k, B_i) - m(C_k, B_j)| \leq |m(C_k, B_i) - m(C_k, B_p)| + |m(C_k, B_p) - m(C_k, B_j)|$$

As real positive numbers interfere in the inequality for which it is true, it yields that property c) exists.

Observations.

a) Distances q_{ij} can be regarded as signifying "the closeness degree" between the databases B_i and B_j, while d_{ij} represents "the remoteness degree" of the two databases

b) In the architecture of a SID, one can decide that the databases that are accessed concomitantly by query processing and situated as "close" as possible according to the definition of the closeness degree mentioned earlier, and whose distance does not exceed a given threshold P can be merged and their clusters can be merged as well.

References

[1] G. Moldovan: Reorganizarea unei baze de date distribuite. Univ. Cluj - Napoca, Fac. Math. and Computer Sci., Res. Sem. Preprint no.5, 1992, 116 -123;

[2] G. Moldovan: O problemă de redistribuire a serviciilor într-un sistem partiţionat în zone distincte după anumite criterii. "Babeş-Bolyai"University Cluj-Napoca, Fac. Math. and Computer Sci., Res. Sem. Preprint no.5, 1993, 5-8

[3] G. Moldovan, M. Văleanu: Redistributing databases in a computer network Analele Univ. Bucureşti, Ser. Math.-Info., 56, 2006

[4] G. Moldovan, M. Văleanu: The Performance Optimization for Date Redistributing System in Computer Network. International Journal of Computers, Communications and Control, vol.I, 2008, Supplementary Issue - Proceedings of ICCCC 2008, Agora Univ., Oradea, 470-473

Mădălina Văleanu, Grigor Moldovan
Babes-Bolyai University
Str. Kogalniceanu,1
400084 Cluj-Napoca, Romania
email:*moldovan@nessie.cs.ubbcluj.ro*