ALLOCATION FRAGMENTS
OF THE DISTRIBUTED DATABASE

The paper describes the distribution fragments of the database under a mathematical model with criterial function involving the influence of the Transaction and Concurrency processing in Database systems. The model could solve variants for replication of the fragments setting constraints in the model. This approach is prepared for revision of actual distribution using real values of the database (cardinality of tables, referential integrity, important requests etc).

1. Introduction

The design of a distributed database system involves making decisions on the placement of data and programs across the sites of a computer network. In distributed database systems the main problem of distribution is the data distribution.

The Database Allocation Problem (DAP) model dates back to the mid-1970s to the work of Eswaran (1974) [Eswaran75], Levin and Morgan (1975) [Levin75], and others. One of the best is described precisely in [Ozsu91]. DAP has been studied in many specialized settings. In 1975 Eswaran [Eswaran75] proved the simple file allocation model as NP-complete. All known solutions of the allocation were solved with heuristic algorithms.

2. Mathematical model

Our model is based on the work of Valduriez and Ozsu [Ozsu91] and teamwork of Jaroslav Pokorný from Charles University [Pokorny92] with enlarged results of the research project in our university.

For an allocation model we need to know: database information, site information, network information and set of constraints. Each of them defines the set of parameters for the allocation model. The cost unit will be a/the time unit.

Database information
We need to know:
The set of fragments, [Matiasko02]
The size of each fragment,
The selectivity of each fragment,
The read access,
The update access,
The read polarization,
The update polarization.

The size of fragment.
The size of the fragment $F_i$ is given by

$$\text{size}(F_i) = \text{card}(F_i) \times \text{length}(F_i)$$

where

- $\text{length}(F_i)$ is the length in bytes of one tuple of fragment $F_i$,
- $\text{card}(F_i)$ is the cardinality of the fragment $F_i$ and it is number of tuples in the fragment.

The selectivity of the fragment

The selectivity of the fragment $F_i$ is given by $\text{sel}(F_i)$ where it is number of tuples of $F_i$ that need to be accessed in order to precede $q_i$.

Read access

Read access $f_{rq}$ is the number read access (frequenting of requests) that the query $q_i$ makes to a fragment $F_j$ during its execution [Matma99a, Matma99b].

Update access

Update access $f_{wu}$ is the number update access (frequenting of requests) that the query $q_i$ makes to a fragment $F_j$ during its execution.

Polarization read access

Polarization read access $r_{ij}$ is the localization the fragments in the query

where

- $r_{ij} = 1$ if the query $q_i$ reads from the fragment $F_j$,
- $r_{ij} = 0$ if the query $q_i$ does not read from the fragment $F_j$.

Polarization update access

Polarization update access $u_{ij}$ is the localization the fragments in the update query

where

- $u_{ij} = 1$ if the query $q_i$ updates the fragment $F_j$,
- $u_{ij} = 0$ if the query $q_i$ does not update the fragment $F_j$.

* Karol Matiaško
Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, E-mail: karol.matiasko@fri.utc.sk
Site information
For each site of the computer network in Slovakia we need to know:
• set of the clients computers \(C_i\) and the set of the queries \(q_i\), running on the these clients’ computers,
• storage capacity,
• processing capacity.

The unit cost of storing data at site \(S_k\) will be \(CM_k\).
The costs of processing one unit of work at site \(S_k\) will be \(CP_k\).
The work unit should be identical with read and update access.

Network information
For the network we need to specify the communication cost.
\(c_g\) denotes the communication cost between site \(S_i\) and \(S_j\). This cost depends on the protocol overhead, distances between sites, channel capacities, etc.

For each query \(q_i\) it is necessary to solve the simple decomposition operation.

Decision variables
The decision variable is \(x_{ij}\), and it is binary.

\[x_{ij} = \begin{cases} 1 & \text{if the fragment } F_j \text{ is stored at site } S_i, \\ 0 & \text{if the fragment } F_j \text{ is not stored at site } S_i \end{cases}\]

Objective function
\[
\text{minimize } N = \sum_{q_i \in Q_i} \text{ND}_{q_i} + \sum_{S_k \in S} \sum_{F_j \in F} \text{NM}_{jkm}.
\]

or
\[
N = \sum_{q_i \in Q_i} \text{ND}_{q_i} \text{ if the memory costs are not important}
\]

where
\(\text{ND}_{q_i}\) is the query processing cost of application \(q_i\),
\(\text{NM}_{jkm}\) is the fragment storing cost of fragment \(F_j\) on the site \(S_k\).

The storage costs are given by
\[
\text{NM}_{jkm} = CM_k \times \text{size}(F_j) \times x_{jk}
\]
and the two summations give us the total storage costs at all sites for all fragments of the computer network.

The query processing costs are given by
\[
\text{ND}_{q_i} = \text{ND}_{Bq_i} + NT_{q_i}
\]

where
\(\text{ND}_{Bq_i}\) is database-processing cost for the application \(q_i\),
\(NT_{q_i}\) is transmission cost for the application \(q_i\).

The processing costs are given by
\[
\text{ND}_{Bq_i} = \text{NRW}_{q_i} + \text{NIC}_{q_i}
\]

where
\(\text{NRW}_{q_i}\) is the access cost for the query \(q_i\) to fragment \(F_j\),
\(\text{NIC}_{q_i}\) is the integrity and concurrency enforcement cost for the query \(q_i\) to fragment \(F_j\).

The access costs are given by
\[
\text{NRW}_{i} = \sum_{S_k \in S} \sum_{F_j \in F} (u_y \times f^*_{ij} + r_y \times f^*_j \times x_{jk} \times \text{CP}_{jk})
\]

The summation gives us the total number of update and read accesses for all fragments referenced by the query \(q_i\). Multiplication by \(\text{CP}_{jk}\) gives us the cost of this access at site \(S_k\).

The \(N\) cost and \(NC\) cost can be specified much like the processing component and depend on the actual computer, operating system, database system and the set of queries performed on the actual site of the computer network.

\[
\text{NIC}_{i} = (\text{KN}_{i} + \text{KN}_j) \times \text{NRW}_{i}
\]

\(\text{KN}_i\) is the integrity enforcement coefficient for the query \(q_i\) to fragment \(F_j\),
\(\text{KN}_i\) is the concurrency enforcement coefficient for the query \(q_i\) to fragment \(F_j\).

\[0 \leq \text{KN}_i \leq 1\]
\[0 \leq \text{KN}_j \leq 1\]

The transmission cost
The transmission costs are different for read and for update access. If the update request exists, it is necessary to make it on all sites where replicas are situated. For read access we need read only one of the copies.

The transmission cost for the query \(q_i\) is given by
\[
\text{NT}_{i} = \text{NTW}_{i} + \text{NTR}_{i}
\]

The update component \(\text{NTW}_i\) of the transmission is
\[
\text{NTW}_i = \sum_{S_k \in S} \sum_{F_j \in F} (f^*_{ij} \times u_y \times x_{jk} \times w_{d(i)(j)}(F_j)) +
\]
\[
+ \sum_{S_k \in S} \sum_{F_j \in F} (f^*_j \times u_y \times x_{jk} \times w_{d(j)(i)}(F_j))
\]

where the first term is sending the update message to the originating site \(i\) of \(q_i\), to all the fragment replicas that need to be updated. The second term is for the confirmation. [Matgr98]

The value \(w_{d}\) is the value of the transmission time for sending the request or answer message from the origin site of the query \(q_i\) to the site \(S_k\).

For \(w_{d}\), we suppose \(w_{d(i)(j)}(F_j) = \text{length}(F_j)/V_{d(i)(j)}\)
\(z(i)\) is the assignment of the origin of the query \(q_i\).

The retrieve component \(\text{NTR}_i\) of the transmission is
\[
\text{NTR}_i = \sum_{F_j \in F} \min_{S_k \in S} (r_y \times x_{jk} \times w_{d(i)(j)}(F_j)) +
\]
\[
+ ((r_y \times x_{jk} \times (\text{sel})(F_j)(\text{size}(F_j))) \times 1/V_{d(i)(j)}
\]
where the first part represents the cost of transmitting the read request to those sites which have copies of fragments that need to be accessed. The second one gives transmission cost for the result of the request.

\[ V_a = \text{transmission velocity from the site } S_i \text{ to the site } S_j \]

For \( a \) we suppose \( w_a(F_j) = \frac{\text{length}(F_j)}{V_a} \)

Constraints

1. **Basic variant** – suboptimal solution with location fragments
2. **Centralized variant** – suboptimal solution with centralized execution time of \( q_i \), \( q_i \in Q \) exist, then

\[ NDB \leq T_i, \quad \forall q_i \in Q \]

The storage constraint

If \( M = \{m_i\}, S_h \subseteq S \) is the set of the storage capacity at each site \( S_h \) then

\[ \sum_{F_i \in F} \text{size}(F_i) * x_{ik} \leq m_i, \quad \forall S_h \subseteq S \]

3. **Experiments**

For the verification of the model we used Greedy Heuristic [Albandoz94], [Francis 89], [Matiaško98] with orientation to the next experiments:

1. Basic variant – suboptimal solution with location fragments without replication
2. Centralized variant – suboptimal solution with centralized variant, when all fragments are localized on the same node
3. Nonfragmented variant – suboptimal solution without fragmentation.
4. Modified variant – suboptimal solution with changing ratio destructive and nondestructive operation for the basic variant

A data model and data of information system of our university were used for the experiments with allocation. For computation as a data sample, data of 20 real applications from the information system of our university were used, which was working on five database relations and fragments allocation to five nodes of the university network. Two of these were used on the remote campuses in Prievidza and Ružomberok, and the others were used within the campus in Žilina.

The sets of fragments \( F = \{F_i\} \) were defined, where particular fragments corresponding with relations or fragments of relations under the following data model:

- **Relation **\( \text{Person} \) is horizontally fragmented by derived fragmentation by joining relation \( \text{Student} \), with a study town to
  - \( F_4 \) is relation \( \text{PersonZA} \)
  - \( F_5 \) is relation \( \text{PersonPD} \)
  - \( F_6 \) is relation \( \text{PersonRB} \)
- **Relation **\( \text{Education} \) is horizontally fragmented by derived fragmentation by joining the relation \( \text{Student} \), with a study town to
  - \( F_7 \) is relation \( \text{EducationZA} \)
  - \( F_8 \) is relation \( \text{EducationPD} \)
  - \( F_9 \) is relation \( \text{EducationRB} \)
- **Relation **\( \text{Course} \) is fragment \( \text{Crepresents static part of database} \)

Applications:

As a set of application \( A = \{a_i\} \) we prepared 10 of the most typical selections and 10 of the most typical destructing operations from our university information system which created an experimental base for verification functionality of allocation for various counted variants.

\[ a_1 \text{ – selection form } F_1 * F_2 * F_3 * F_4 * F_5 * F_6 \]
\[ a_2 \text{ – selection form } F_2 * F_3 * F_4 * F_5 * F_6 \]
\[ a_3 \text{ – selection form } F_3 * F_4 * F_5 * F_6 \]
\[ a_4 \text{ – selection form } F_4 * F_5 * F_6 \]
\[ a_5 \text{ – selection form } F_5 * F_6 \]
\[ a_6 \text{ – selection form } F_6 \]
\[ a_7 \text{ – selection form } F_1 \otimes F_2 \otimes F_3 \]
\[ a_8 \text{ – selection form } F_1 \otimes F_2 \otimes F_3 \]
\[ a_9 \text{ – selection form } F_1 \otimes F_2 \otimes F_3 \]
\[ a_{10} \text{ – selection form } F_{10} \]
\[ a_{11} \text{ – } a_{10} \text{ update in the fragments } F_1 \text{ to } F_{10} \]

where \( \otimes \) is operation UNION.

The values of monitored features were measured during a normal running of the information system. These features represented frequentations of nondestructive operations, selection of particular fragments, response times between a workplace of the network, size of relations of particular fragments and duration of elementary operations.
First experiment presents the basic variant. The main goal is searching the suboptimal solution of the one level fragmentation. One-level fragmentation means that each fragment will be used only one time. The best allocation of the fragments is illustrated in Fig. 1.

The objective function for this variant has the value of 878202. This result shows that most fragments are allocated to the workplaces, which provides minimal cost considering transmission speed in the network.

We prepared an intuitive allocation, which related with the method BestFeed [Ceri84]. In this variant every fragment is situated to that workplace, under its maximal query frequency. If we suppose no destructive operation, the objective function enhances to the value 783035 and another fragment allocation – Fig. 2.

From the point of view of destructive operation (DELETE, INSERT, UPDATE), then optimal allocation is another – Tab. 1, and objective function has the value of 362417. It is important and interesting to watch the influence of destructive operations to behavior of the whole system.

During experiments the centralized variant was made. Experiments with all allocated fragment are always on the same node. For each node we get one variant of the solution. The results are in the Tab. 2.

According to the results the centralized variant would be the best as allocated fragments on the node S4 with objective function value 953792.

Table of the costs with real and percentage deviation form optimum (N – cost, DN – difference cost of optimal value, % – difference cost of optimal value)

| Variant   | N      | DN    | %     |
|-----------|--------|-------|-------|
| Variant1  | 878202 | 0     | 0     |
| Variant2  | 2077116| 1198914| 57   |
| Variant3  | 1754237| 876035| 49    |
| Variant4  | 2624590| 1746388| 66   |
| Variant5  | 953792 | 75590 | 7     |
| Variant6  | 1026311| 148109| 14    |

When we compare the result, which we get for the fragmental variant, it is different from the optimal value by 12 percent.

When we treat the nonfragmented variant, in which the fragments F1, F2, F3 collect one fragment, allocated always on the one node, and by the same way fragments F4, F5, F6 and fragments F7, F8, F9 then the cost for distribution has the value of the objective function 1000908 – (Tab.3).

When we compare the results of the fragmental and nonfragmented variants, we can see the differences in cost. The nonfragmented variant costs 12 percent more than the fragmental variant.

Fig. 4 Comparison of fragmented and nonfragmented variants

| Variant | N      | DN    | %     |
|---------|--------|-------|-------|
| Nonfragmented variant | 1000908| 122706| 12q   |
Change of the cost when the number of the “select” is constant

| Variant | N1       | DN       | %  | Destructive operation [%] |
|---------|----------|----------|----|---------------------------|
| 8       | 1291932  | 413730   | 32 | 100                       |
| 9       | 1235437  | 357235   | 28 | 90                        |
| 10      | 1178942  | 300740   | 25 | 80                        |
| 11      | 1129705  | 251503   | 22 | 70                        |
| 12      | 1065964  | 187762   | 17 | 60                        |
| 13      | 1009473  | 131717   | 13 | 50                        |
| 14      | 952984   | 74782    | 7  | 40                        |
| 15      | 896491   | 18289    | 2  | 30                        |
| 16      | 840000   | -38202   | -5 | 20                        |
| 17      | 792581   | -85621   | -11| 10                        |

Change of the cost when the number of the “update” is constant

| Variant | N2       | DN       | %  | Nondestructive operat. [%] |
|---------|----------|----------|----|---------------------------|
| 18      | 1277275  | 399073   | 31 | 100                       |
| 19      | 1206130  | 327928   | 27 | 90                        |
| 20      | 1134989  | 256787   | 22 | 80                        |
| 21      | 1075923  | 197721   | 18 | 70                        |
| 22      | 992704   | 114502   | 11 | 60                        |
| 23      | 921562   | 43360    | 4  | 50                        |
| 24      | 850418   | -27784   | -4 | 40                        |
| 25      | 779275   | -98927   | -13| 30                        |
| 26      | 708133   | -170069  | -25| 20                        |
| 27      | 636991   | -241211  | -38| 10                        |

From the point of view of the modified variant, we compared two situations. In the first variant we watched how the value of the objective function is changed (N1) when the number of the selected operation (only SELECT) is constant, and the number of the destructive operation is changed. At the beginning of this experiment the frequencies of all the kinds of operation are the same. On the next variant the number of the destructive operations is reduced by 10 percent. The objective function is improved by 30 percent of the number of destructive operations. DN is difference of the cost for the variant and optimal.

In another case of this variant we watched the change of the value of the objective function (N2) when the number of the destructive operations is constant and the number of the nondestructive is changed, as in the previous variant, in each step by 10 percent. The objective function value is improved by 50 percent of the number of nondestructive operations. DN is the difference between variant costs and optimal costs.

References

[1] ALBANDOZ, J. P. e col.: Lecturas en Teoria de Localization, Universidad de Sevilla, 1996.
[2] CERI, S., PELAGATTI, G.: A solution method for the Allocation problem in Distributed Databases, Process Letters, 10, 1982.
[3] ESWARAN, K., P.: Placment of records in a file and file allocation in a computer network, Information Processing 1974: pp. 304-307, North Holland Publ. Co., Amsterdam, 1974.
[4] ESWARAN: The notions of consistency and predicate locks in a Database Systems, CACM 11,1975.
[5] FRANCIS, R. L., MIRCHANDANI, P.: Discrete location theory, Wiley, New York 1989.
[6] LEVIN, K. D. and MORGAN, H. L.: Optimizing distributed databases- A Framework for research, Proc. AFIPS NCC 1975, pp. 473-478, AFIPS Press, 1975.
[7] MATIAŠKO, K.: Distributed database modeling (in slovak), Žilinska univerzita, 1998, Žilina.
[8] OZSU, VALDURIEZ: Principles of Distributed Database Systems, Prentice Hall,Englewood Cliffs, New Jersey 1991
[9] POKORNY, J., SOKOLOWSKY, P., PETERKA, J.: Distributed database systems (in czech), Academia, Praha 1992.
[10] MATIAŠKO, K.: Database systems (in slovak), 2002, EDIS Žilina, [11] MATIAŠKO, K., GRONDŽÁKOVÁ, E.: Data Migration, Studies of the Faculty of Management Science 7, 1998
[12] MATIAŠKO K. MARTINCOVÁ: Principles of Query Processing decomposition and parallel scheduling of the execution Part I., Query decomposition, Studies 8/1999, Faculty of Management Science and Informatics, 1999
[13] MATIAŠKO K. MARTINCOVÁ: Principles of Query Processing decomposition and parallel scheduling of the execution Part II., Parallel scheduling of the query execution, Studies 8/1999, Faculty of Management Science and Informatics, 1999.