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 [Pokorný92] 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, \[\text{size} (F_j) = \text{card}(F_j) \times \text{length}(F_j)\]

where

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

The selectivity of the fragment

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

Read access

Read access \(f_{ri}\) 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 \(f_{ui}\) 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 \(F_j\).
Site information
For each site of the computer network in Slovakia we need to know:
• set of the clients computers \( C_k \) and the set of the queries \( q_i \) running on 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. \( c_{ij} \) 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.

\( x_{ij} = 1 \) if the fragment \( F_j \) is stored at site \( S_i \)
\( x_{ij} = 0 \) if the fragment \( F_j \) is not stored at site \( S_i \)

Objective function
\[
\text{minimize} \quad N = \sum_{q_i \in Q} N_{D_i} + \sum_{S_j \in S} \sum_{F_j \in F} N_{M_{jk}},
\]
or
\[
N = \sum_{q_i \in Q} N_{D_i} \quad \text{if the memory costs are not important}
\]
where
\( N_{D_i} \) is the query processing cost of application \( q_i \)
\( N_{M_{jk}} \) is the fragment storing cost of fragment \( F_j \) on the site \( S_k \)

The storage costs are given by
\[
N_{M_{jk}} = CM_k \cdot \text{size}(F_j) \cdot x_{jk}
\]
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
\[
N_{D_i} = N_{DB_i} + N_{T_i}
\]
where
\( N_{DB_i} \) is database-processing cost for the application \( q_i \)
\( N_{T_i} \) is transmission cost for the application \( q_i \)

The processing costs are given by
\[
N_{DB_i} = N_{RW_i} + N_{IC_i}
\]
where
\( N_{RW_i} \) is the access cost for the query \( q_i \) to fragment \( F_j \)
\( N_{IC_i} \) is the integrity and concurrency enforcement cost for the query \( q_i \) to fragment \( F_j \)

The access costs are given by
\[
N_{RW_i} = \sum_{S_j \in S} \sum_{F_j \in F} (w_{ij} \cdot x_{jk} \cdot w_{d(i,k)}(F_j))
\]
The summation gives us the total number of update and read accesses for all fragments referenced by the query \( q_i \). Multiplication by \( CP_k \) gives us the cost of this access at site \( S_k \).

The \( N_T \) cost and \( N_C \) 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.

\[
N_{IC_i} = (KNI_i + KNC_i) \cdot N_{RW_i}
\]
\( KNI_i \) is the integrity enforcement coefficient for the query \( q_i \) to fragment \( F_j \)
\( KNC_i \) is the concurrency coefficient for the query \( q_i \) to fragment \( F_j \)

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
\[
N_T = N_{TW_i} + N_{TR_i}
\]
The update component \( N_{TW_i} \) of the transmission is
\[
N_{TW_i} = \sum_{S_j \in S} \sum_{F_j \in F} (f_{ij} \cdot u_{ij} \cdot x_{jk} \cdot w_{d(i,k)}(F_j)) +
+ \sum_{S_j \in S} \sum_{F_j \in F} (f_{ij} \cdot u_{ij} \cdot x_{jk} \cdot w_{d(i,k)}(F_j))
\]
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 \( w_{ij} \) 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 \( w_{d(i,k)} \) we suppose \( w_{d(i,k)}(F_j) = \text{length}(F_j)/V_{d(i,k)}(z(i)) \) is the assignment the origin of the query \( q_i \) to the site \( S_k \).

The retrieve component \( N_{TR_i} \) of the transmission is
\[
N_{TR_i} = \sum_{F_j \in F} \min \left( r_{ij} \cdot x_{jk} \cdot w_{d(i,k)}(F_j) \right) +
+ ((r_{ij} \cdot x_{jk} \cdot (sel)(F_j)/size(F_j))) \cdot 1/V_{d(i,k)}(z(i))
\]
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.

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

For \( w_{ik} \) we suppose \( w_{ik}(F_j) = \text{length}(F_j)/V_{ik} \)

Constraints

The response time constraint

Let the set \( T = \{ T^j_i \} \) of the maximum response time of \( q_i \), \( q_i \in Q \) exist, then

\[ NDB_i \leq T^j_i, \quad \forall q_i \in Q \]

execution time of \( q_i \) is less equal than maximum response time of \( q_i \)

The storage constraint

If \( M = \{ m_k \}, \text{Sk} \Delta S \) is the set of the storage capacity at each site \( S_k \) then

\[ \sum_{F_j \in F} \text{size}(F_j) \times x_{jk} \leq m_k, \quad \forall S_k \in S \]

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
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 Prešov and Ružomberok, and the others were used within the campus in Žilina.

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

- Relation `Person` is horizontally fragmented by derived fragmentation by joining relation `Student`, with a study town to
- \( F_4 \) is relation `PersonZA`
- \( F_5 \) is relation `PersonPD`
- \( F_6 \) is relation `PersonRB`
- Relation `Education` is horizontally fragmented by derived fragmentation by joining the relation `Student`, with a study town to
- \( F_7 \) is relation `EducationZA`
- \( F_8 \) is relation `EducationPD`
- \( F_9 \) is relation `EducationRB`
- Relation `Course` is fragment `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.

\[ a_{1} - \text{selection form } F_1 \ast F_2 \ast F_3 \ast F_4 \ast F_5 \ast F_6 \]
\[ a_{2} - \text{selection form } F_2 \ast F_3 \ast F_4 \ast F_5 \ast F_6 \ast F_7 \]
\[ a_{3} - \text{selection form } F_1 \ast F_2 \ast F_3 \ast F_4 \ast F_5 \ast F_6 \]
\[ a_{4} = a_{1} \otimes a_{2} \otimes a_{3}, \]
\[ a_{5} - \text{selection form } F_1 \otimes F_2 \otimes F_3 \]
\[ a_{6} - \text{selection form } F_1 \otimes F_2 \otimes F_3 \otimes F_4 \]
\[ a_{7} - \text{selection form } F_1 \otimes F_2 \otimes F_3 \otimes F_4 \]
\[ a_{8} - \text{selection form } F_1 \ast F_2 \ast F_3 \ast F_4 \ast F_5 \]
\[ a_{9} - \text{selection form } F_1 \ast F_2 \ast F_3 \ast F_4 \ast F_5 \ast F_6 \]
\[ a_{10} - \text{selection form } F_1 \ast F_2 \ast F_3 \ast F_4 \ast F_5 \ast F_6 \ast F_7 \]
\[ a_{11} - a_{20} \text{ update in the fragments } F_1 \text{ to } F_{10} \]

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.

![Fig. 1 Allocation of the fragments with one-level replications](image-url)
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 $S_4$ with objective function value 953792.
Change of the cost when the number of the “select” is constant  
Tab. 4

<table>
<thead>
<tr>
<th>Variant</th>
<th>N1</th>
<th>DN</th>
<th>% Destructive operation [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1291932</td>
<td>413730</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1235437</td>
<td>357235</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>1178942</td>
<td>300740</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>1129705</td>
<td>251503</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>1069696</td>
<td>187762</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>100473</td>
<td>131271</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>952984</td>
<td>74782</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>896491</td>
<td>18289</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>840000</td>
<td>-38202</td>
<td>-3</td>
</tr>
<tr>
<td>17</td>
<td>792581</td>
<td>-85621</td>
<td>-11</td>
</tr>
</tbody>
</table>

Change of the cost when the number of the “update” is constant  
Tab. 5

<table>
<thead>
<tr>
<th>Variant</th>
<th>N2</th>
<th>DN</th>
<th>% Nondestructive operat. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>1277275</td>
<td>399073</td>
<td>31</td>
</tr>
<tr>
<td>19</td>
<td>1206130</td>
<td>327928</td>
<td>27</td>
</tr>
<tr>
<td>20</td>
<td>1134989</td>
<td>256787</td>
<td>22</td>
</tr>
<tr>
<td>21</td>
<td>1075923</td>
<td>197721</td>
<td>18</td>
</tr>
<tr>
<td>22</td>
<td>992704</td>
<td>114502</td>
<td>11</td>
</tr>
<tr>
<td>23</td>
<td>921562</td>
<td>43360</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>850418</td>
<td>-27784</td>
<td>-4</td>
</tr>
<tr>
<td>25</td>
<td>779275</td>
<td>-98927</td>
<td>-13</td>
</tr>
<tr>
<td>26</td>
<td>708133</td>
<td>-170069</td>
<td>-25</td>
</tr>
<tr>
<td>27</td>
<td>636991</td>
<td>-241211</td>
<td>-38</td>
</tr>
</tbody>
</table>

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.

4. Conclusion

Development of information technology allows development of information systems effectively and in harmony with organization structure of firms. Therefore, distributed database systems are the tools that are helpful for the development of those systems. But designing the data model for a distributing database system is always challenge from the fragmentation database to the allocation the fragments or all databases, regardless of the available conditions.

References

[1] ALBANDOZ, J. P. e col.: Lecturas en Teoria de Localization, Universidad de Sevilla, 1996,
[4] ESWARAN: The notions of consistency and predicate locks in a Database Systems, CACM 11,1975,
[10] MATIAŠKO, K.: Database systems (in slovak), 2002, EDIS Žilina,