ALLOCATION DATA FRAGMENTS FOR THE DISTRIBUTED DATABASE

Karol Matiaško and Jana Hamalová

Department of Informatics, Faculty of Management Science and Informatics, University of Zilina, Moyzesova 20, 01026, Zilina, SLOVAKIA, karol.matiasko@fri.utc.sk, jana.hamalova@fri.utc.sk

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) [1], Levin and Morgan (1975) [2], and others. The common one is described precisely in [3]. DAP has been studied in many specialized settings. In 1975 Eswaran [1] proved the simple file allocation model as NP-complete. All known solutions of the allocation were solved with heuristic algorithms.

2. Mathematics Model

Our model is based on the work of Valduriez and Ozsu [3] and teamwork of Jaroslav Pokorný from Charlie's University [4] 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 know:

- the set of fragments,
- 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_j is given by

- $size(F_j)=card(F_j)\times length(F_j)$,where $length(F_j)$ is the length in bytes of one tuple of fragment F_j,
- $card(F_j)$ 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 $sel_i(F_j)$ 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^r_{ij} is the number read access (frequenting of requests) that the query q_i makes to a fragment F_j during its execution.

Update access

Update access f^w_{ij} 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 doesn't 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 S_k of computer network we need to know:

- set of the clients computers C_{jk} 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_{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

Minimize $N=\sum_{\forall q_i \in Q_i} ND_i + \sum_{\forall S_k \in S} \sum_{\forall F_j \in F} NM_{jk}$ or minimaze $N=\sum_{\forall q_i \in Q_i} ND_i$ if the memory costs are not important. Where ND_i is the query processing cost of application q_i; NM_{jk} is the fragment storing cost of fragment F_j on the site S_k.

The storage costs are given by

$$NM_{jk}=CM_k \times size(F_j) \times x_{jk}$$

and the two summations give us the total storage costs at all sites for all fragments of the computer network.

The query processing cost are given by

$$ND_i=NDB_i+NT_i,$$

where NDB_i is database-processing cost for the application q_i; NT_i is transmission cost for the application q_i.

The processing costs are given by

$$NDB_i=NRW_i+NIC_i,$$

Where NRW_i is the access cost for the query q_i to fragment F_j NIC_i is the integrity and concurrency enforcement cost for the query q_i to fragment F_j.

The access cost are given by

$$NRW_i=\sum_{\forall S_k \in S} \sum_{\forall F_j \in F} (u_{ij} * f^{w}_{ij} + r_{ij} * f^{r}_{ij}) * x_{jk} * CP_k$$

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 NI cost and NC cost can be specified much like the processing component and depends from the actual computer, operating system, database system and the set of queries performed on the actual site of the computer network.

$$NIC_i=(KNI_i+KNC_i) \times NRW_i,$$

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.

$$0 \le KNI_i \le 1 \quad \text{and} \quad 0 \le KNC_i \le 1$$

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

$$NT_i=NTW_i+NTR_i.$$

The update component NTW_i of the transmission is

$$NTW_i=\sum_{\forall S_k \in S} \sum_{\forall F_j \in F} (f^{w}_{ij} * u_{ij} * x_{jk} * w_{z(i),k}(F_j)) + \sum_{\forall S_k \in S} \sum_{\forall F_j \in F} (f^{w}_{ij} * u_{ij} * x_{jk} * w_{k,z(i)}(F_j))$$

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.

The value $w_{i,k}$ is the value of the transmission time for sending the request or answer message from origin site of the query q_i to the site S_k.

For $w_{z(i),k}$ we suppose $w_{z(i),k}$-$k(F_j)=length(F_j)/V_{z(i),k}$ $z(i)$ is the assignment the origin of the query q_i.

The retrieve component NTR_i of the transmission is

$$NTR_i=\sum_{\forall F_j \in F} f^{r}_{ij} \min_{\forall S_k \in S}(r_{ij} * x_{jk} * w_{z(i),k}(F_j)) + ((r_{ij} * x_{jk} * (sel_i(F_j)/fsize(F_j)))) * 1/V_{z(i),k}$$

where the first part represents the cost of transmitting the read request to those sites, which have copies of fragments that need be accessed. The second one gives transmission cost for the result of the request.

V_{ij} is the transmission velocity from the site S to the site S_j. For w_{ik} we suppose $w_{ik}(F_j)=length(F_j)/V_{ik}$.

Constraints

The response time constraint

Let exist the set $T=\{T_i^Q\}$ of the maximum response time of q_i, $q_i \in Q$ then

$$NDB_i \le T_i^Q, \quad \forall q_i \in Q,$$

execution time of q_i is less equal than maximum response time of q_i

The storage constraint

If $M=\{m_k\}$, $S_k \in S$ is the set of the storage capacity at each site S_k then

$$\sum_{\forall F_j \in F} size(F_j) \times x_{jk} \le m_k, \qquad \forall S_k \in S.$$

3. EXPERIMENTS

For the verification of the model was used Greedy Heuristic [5-7] 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. Modificated 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 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 on the campus in Žilina.

There were defined sets of fragments $F=\{F_i\}$, where particular fragments corresponding with relations or fragments of relations under following data model:

- Relation **Student** is horizontally fragmented by study town to
  - F_1 is relation StudentZA
  - F_2 is relation StudentPD
  - F_3 is relation StudentRB
- Relation **Person** is horizontally fragmented by derived fragmentation by joining with relation Student, by 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 with relation Student, by 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 prepare 10 of the most typical selections and 10 of the most typical destructing operations from our university information system, which made an experimental base for verification functionality of allocation for various counted variants.

- a_1 -selection form $F_{1*} * F_{4*} * F_7 * F_{10}$
- a_2 - selection form $F_{2*} * F_5 * F_8 * F_{10}$
- a_3 - selection form $F_{3*} * F_6 * F_9 * F_{10}$
- $a_4 = a_1 \otimes a_2 \otimes a_3$
- a_5 - selection form $F_1 \otimes F_2 \otimes F_3$
- a_6 - selection form $F_4 \otimes F_5 \otimes F_6$
- a_7 - selection form $F_3 \otimes F_7 \otimes F_9$
- a_8 - selection form $F_1 * F_4 \otimes F_{2*} * F_5 \otimes F_{3*} * F_6$
- a_9 - selection form $F_7 * F_{10} \otimes F_8 * F_{10} \otimes F_9 * F_{10}$
- a_{10} - selection form F_{10}
- a_{11} - a_{20} update in the fragments F_1 to F_{10} ,

where $\otimes$ is operation UNION.

The values of monitored features we measured during a normal running of the information system. These features represented frequentations of nondestructive operations, selection of particular fragments, response times between workplace of the network, size of relations of particular fragments and making time of elementary operations.

As a first experiment were made solution of basic variant, searching for 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.

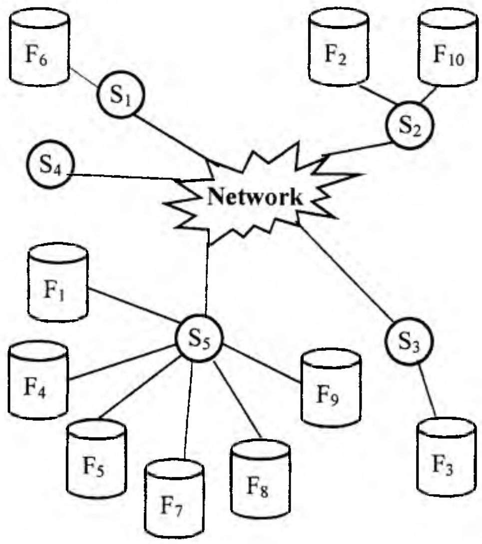

*Figure*1. Allocation of the fragments with one-level replications

The objective function for this variant has value 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 method BestFeed [8] where every fragment allocated to that workplace, from its maximal query frequency. If we suppose no destructive operation, the objective function enhances to the value 783035 and another fragment allocation (see fig. 2).

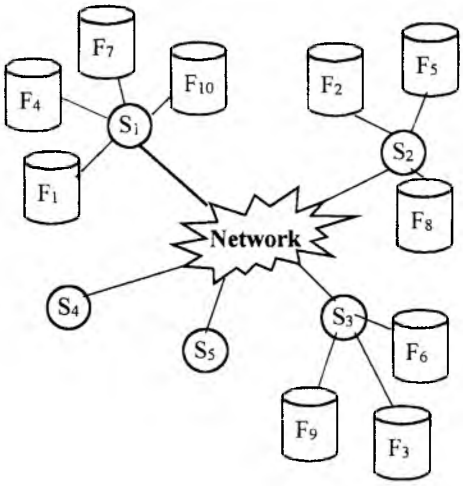

Figure 2. Intuitive Fragments Allocation

When we research only an evolution destructive operation (DELETE, INSERT, UPDATE), then optimal allocation is another - tab. 1, and objective function has value 362417. It is important and interesting in regard to the impact of destructive operations to running all the systems.

Table 1: Allocation fragment only for the destructive operations

362417	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
S1	0	0	0	0	0	1	0	0	0	0
S2	0	1	0	0	0	0	0	0	0	1
S3	0	0	1	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	0
S5	1	0	0	1	1	0	1	1	1	0

During results the **centralized variant** was made. Experiments with all allocated fragment are always on the same node. For **every** node we get one variant of the solution. The results are in the tab. 2 and fig.3.

Table 2: Table of the costs with real and percentile declamation form optimum (N - cost, DN - difference cost of optimal value, % - difference cost of optimal value)

	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

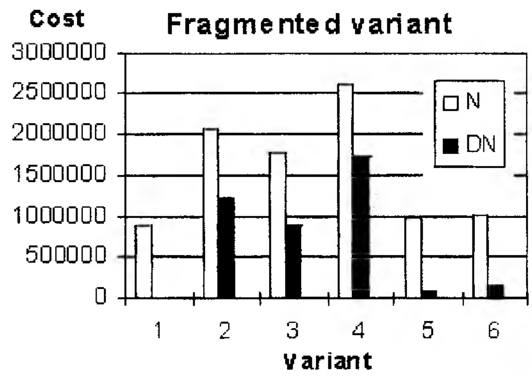

Figure 3. Cost graph for every variant of the solution

According to the results the centralized variant would be the best as allocated fragments on the node S_4 with objective function value 953792.

When we research **the nonfragmented variant**, in which the fragments F_1, F_2, F_3 collect one fragment, allocated always on the one node, and by the same way fragments F_4, F_5, F_6 and fragments F_7, F_8, F_9 then the cost for distributing made the seventh variant with value of the objective function 1000908 – tab.3 and fig.4.

Table 3: The result for the nonfragmented variant

	N	DN	%
Nonfragmented variant	1000908	122706	12

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

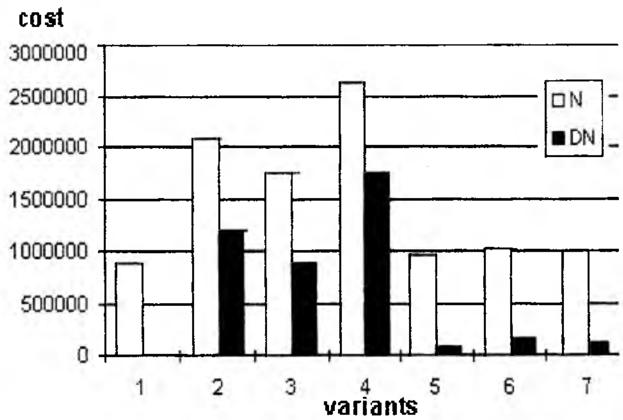

Figure 4. Comparison of fragmented and nonfragmented variants

By result of the **modified variant**, we researched two situations. For the first time we research 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 reduced by 10percent. The objective function is improved by 30percent of the number of destructive operations. DN is difference of the cost for the variant and optimal.

Table 4. 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	131271	13	50
14	952984	74782	7	40
15	896491	18289	2	30
16	840000	-38202	-5	20
17	792581	-85621	-11	10

Table 5. Change of the cost when the number of the "update" is constant

Variant	N2	DN	%	Nondestructive operations v %
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

In another case of this variant we research changing 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 the every step by 10 percent. The objective function value is improved by 50 percent of the number of nondestructive operations. DN is the difference of the cost for the variant and optimal

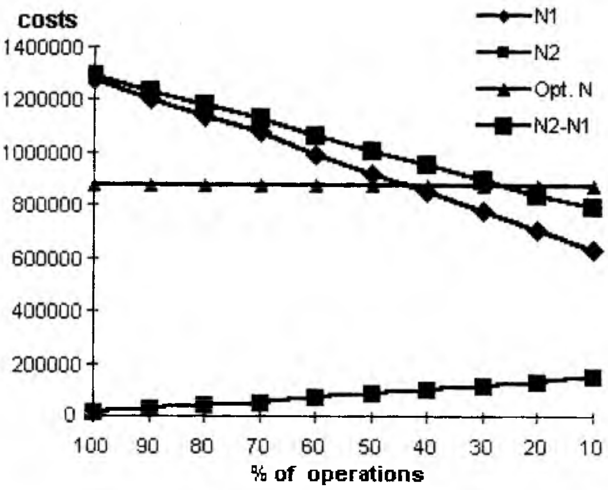

Figure 5. Comparing the costs when the number of operations are changed SELECT and UPDATE

CONCLUSION

Development 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 of 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]. Eswaran, The notions of consistency and predicate locks in a Database Systems, CACM, **11**, 1975.

[2]. Levin K.D. and H.L. Morgan, Optimizing distributed databases- A Framework for research, *Proc. AFIPS NCC*, 1975, pp.473-478.

[3]. Ozsu, Valduriez, Principles of Distributed Database Systems, *Prentice Hall, Englewood Cliffs*, New Jersey 1991.

[4]. Pokorný J.,Sokolowsky P., Peterka J., Distribuované databázové systémy, *Academia*, Praha, 1992.

[5]. Albandoz J.P. e col., Lecturas en Teoria de Localization, *Universidad de Sevilla*, 1996.

[6]. Francis, R.L. and P.Mirchandani, Discrete location theory, *Wiley*, New York, 1989.

[7]. Matiaško K., Modelovanie distribuovaných databáz, *Žilinská Univerzita*, 1998, Žilina.

[8]. Ceri S., Pelagatti G., A solution method for the Allocation problem in Distributed Databases, *Process Letters*, **10**, 1982.