File Redundancy Issues in Distributed Database Systems

Shojiro MURO[†], Toshihide IBARAKI[††], Hidehiro MIYAJIMA[†††] and Toshiharu HASEGAWA[†]

† Dept. of Applied Math. & Physics, Faculty of Engineering, Kyoto University, Kyoto, 606 Japan
†† Dept. of Information & Computer Sciences, Toyohashi University of Technology, Toyohashi, 440 Japan
††† the 2nd Section, the 2nd Dept., Automation System Laboratory, FUNUC Ltd., Hino, 191 Japan

Abstract

This paper treats the file redundancy issue in distributed database systems asking what is the optimal number of file copies, given the ratio *r* of the frequencies of query and update requests. To draw a general conclusion applicable to a wide variety of practical distributed database systems, simplified network models are constructed, and optimal number of file copies, as well as their locations, to minimize the communication cost is computed. By examining various network types, we plot the optimal number of file copies as a function of the ratio *r*.

Our conclusion is that a single copy suffices under moderate condition, and it is disadvantageous to have more than a few copies unless the frequency of query requests is unduly higher (*e.g.*, 50 times) than that of update requests.

1. Introduction

In many papers discussing concurrency control mechanisms and performance of the distributed database systems (abbreviated to DDBS), the full redundancy of the data files is often assumed (i.e., each site in the system has a complete copy of every database (file)). Although it is theoretically easy to treat the systems under the full redundancy assumption, this assumption should be carefully justified from the view point of the system performance. Our conclusion of this paper is quite contrary; a single copy of the file suffices in most of the practical cases and the full redundancy cannot be justified unless the frequency of query requests is extremely higher than that of update requests.

The optimal file redundancy and allocation problem in DDBS has been discussed in a number of papers (e.g., see a survey by Dowdy, et al. [1]). However, most of these papers emphasize the precise formulation of given particular DDBS models. Although these are useful to determine the optimal file redundancy and allocation of a given particular system, the computed results may not be general enough to suggest the characteristics common to many practical DDBS models.

To discuss the system performance in a general manner, the operating costs of DDBS are classified into storage costs of files and communication costs for queries and updates. The high redundancy tends to decrease the communication costs for queries, but increase the storage costs and the communication costs for updates. The ratio *r* of the frequencies of update requests to query requests is a key parameter to determine the optimal redundancy of a given system. We further place the following symplifying assumptions so that optimal redundancy and allocation of files may be easily determined without losing essential characteristics of the systems.

(1) Storage costs are ignored from the operating costs.

(2) The system has only one file, copies of which may be allocated to any sites in the system.

(3) There is no restriction on the communication link capacity.

Note that these assumptions all have the tendency to bias the problem in favor of increasing the number of file copies in the network. This is clear in the case of assumption (1). If there is more than one file and/or the communication links have capacities, the conflicts to access different files through the same communication links must be resolved from the view point of DDBS concurrency control. This will introduce the communication overhead and increase the cost. Thus assumptions (2) and (3) result in an underestimate of the communication cost and hence it favors the high redundancy of the files. Of course, our conclusion would also apply to the systems which have more than one database file, if the above bias is taken into account.

The consistency issue among distributed copies is not also considered in our model. This again biases the problem in the direction of favoring high redundancy. Finally, the reliability attained by having more than one copy is not taken into account, because the cost is solely determined by the communication cost. Only this simplification has the opposite effect of discouraging high redundancy.

As a whole, it seems that the above simplifications tend to give a conclusion overestimating the optimal number of file copies.

The optimal file redundancy and allocation of the above model is formulated as a special case of the simple plant location problem for which reasonably efficient algorithms are known [2], in spite of the fact that it is generally NP-complete. Optimal redundancy of files as well as their allocation is then computed for each of the problem instance, with r, the ratio of the frequencies of update requests to query requests, as a key parameter.

The total of 80 network models, based on three different types of computer communication networks, are randomly generated and solved. As will be described in Section 4, the results for 80 problems have a considerable similarity, suggesting the generality of our results. An important observation is that the optimal number of copies is always 1 if ratio r is more than or equal to 0.23 even in the worst case. In view of the fact that our model tends to favor high redundancy, this would suggest that allocating a large number of copies (i.e., high redundancy) is an unreasonable policy excepting the case in which r is unduly low.

2. Communication Cost and Optimal Redundancy

Consider a communication network with N sites, each of which has a computer with processing and storage capacities. The communication cost of a unit of data through the link (i,j) connecting sites i and j is $c(i,j)$. Note that $c(i,j)=c(j,i)$ is assumed. If a unit of data is transmitted through a channel consisting of links (i_1,i_2), (i_2,i_3), ..., (i_{m-1},i_m), then the communication cost is determined as the sum of all link costs,

$$\sum_{j=1}^{m-1} c(i_j,i_{j+1}).$$

As noted in Introduction, only one type of file is considered but its copy may be stored at any of N sites. Two types of transactions, queries and updates, are issued from the users. When a query transaction is issued from a site, the site determines the channel through which a destination site storing a copy of the file is reached. The query is sent through the channel, reads the copy and is sent back to the original site. When an update transaction is issued at a site, it is sent to all sites storing the copies. For simplicity, we assume that a unit of data is always transmitted through a channel, at each query or update. Let $p(\geqq 1)$ be the number of copies and L_p be a set of the sites storing a copy. Then the channels through which data are sent are determined as follows.

(a) The query request from site i is sent to a site in L_p through the channel with the minimum communication cost. This cost is denoted by $c_q(i,L_p)$.

(b) The update request from site i is sent to every site j in L_p through the channel with the minimum communication cost.

The first problem is to decide the set L_p so that the average query cost

$$\sum_{i=1}^{N} c_q(i,L_p)/N$$

is minimized, assuming for simplicity that queries are issued from all sites with equiprobability. The attained minimum cost is denoted by $C_R(p)$. As is discussed in [3], the value $C_R(p)$ is calculated by applying the algorithm for the simple plant location problem [2].

Let C_U denote the average of the minimum communication cost between a pair of sites in the network. The above assumption (b) implies that the communication cost of updating p copies is approximately given on the average by $p.C_U$. This is not exact because optimal locations are not evenly distributed in the network. But this approximation seems to be quite accurate, as also confirmed in the experiment in Section 4. C_U can be calculated efficiently by applying a shortest path algorithm for all pairs of nodes (e.g., Warshall-Floyd method).

Based on the above argument, the average communication cost per one request of query or update, for the given number of copies p and the ratio r, is given as follows.

$$C(p,r)=(C_R(p)+r.p.C_U)/(1+r).$$

The optimal number of copies (i.e., redundancy) for the given r is defined to be p which minimizes $C(p,r)$, and is denoted $F(r)$.

3. Communication Network Models

Three types of communication network models are randomly constructed for evaluation. In each model, location of N sites, the pairs of sites between which the communication links are laid down, and the associated communication costs are randomly generated.

Type I: N sites are placed in the two dimensional plane by determining their x and y coordinates by uniform random numbers. The links are then randomly chosen so that $d(\%)$ of the all pairs (i,j) have links. Finally the communication cost $c(i,j)$ of link (i,j) is given by Euclidean distance between sites i and j.

Type II: This is different from type I in that pairs (i,j) are connected by links if and only if their distances are smaller than a given threshold value.

Type III: This is different from type I in that the communication costs of links $c(i,j)$ are given by uniform random number.

Type I and type II reflect the reality in the sense that the communication cost $c(i,j)$ is proportional to the distance of sites i and j. Type III is introduced for the reference purpose, since type III does not have a special structure as in types I and II. If the computational results for type III model is similar to those obtained from types I and II, conclusions drawn from the obtained results would be general and can be applied to a wide variety of networks.

4. Computational Results

In our computational experiment, the total of 80 test problems consisting of 16 basic network models are are randomly generated and solved. The $F(r)$ computed by experiment is plotted in Fig.1 for the type I model with N=50 and d=50%, where each point represents the average of 5 test problems. $F(r)$ is a monotone decreasing function in r as easily inferred by the intuitive understanding of the system. Other types of problems all have similar tendency in spite of the differences in types, N values and d values. In particular, the plotted $F(r)$ decreases from 2 to 1 as r becomes larger than a value ranging from 0.14 to 0.23. These critical r values are listed in Table 1. It is interesting to see that this value seems to be almost independent of problem types, sizes N and densities d, except that type II tends to have slightly larger values than others. As stated in Section 3, this suggests the generality of the obtained results.

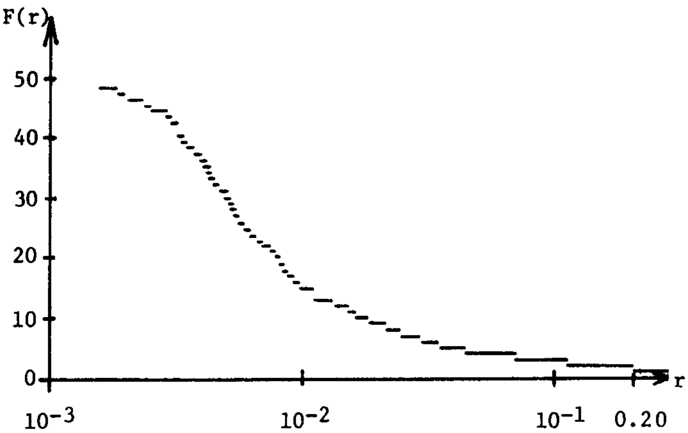

Fig. 1. F(r) vs. r for the type I network with N = 50 and d = 50(%)

5. Concluding Remarks

In view of the fact that our formulation is biased in the direction of favoring high redundancy, the computational results strongly suggest that a single copy suffices under moderate condition (e.g., the query requests are issued at most 4~7 times more than the update requests), and it is disadvantageous to have more than a few copies unless the ratio r of the frequencies of the update requests to the query requests is unduly low (e.g., less than 0.02).

In database systems in the real world, the ratio r varies from application to application. For example, database systems of banks would have the ratio r which is considerably larger than 1, and database systems of travel agencies would have the ratio r approximately equal to 1 (according to the data provided by Japan Travel Bureau). On the other hand, in the literature reference database systems of libraries, almost all requests are queries, implying that r is very small. Our conclusion stated above suggests that the database systems of banks and travel agencies should install a single copy of each file, while the database systems of libraries may benefit from having a number of copies allocated in various sites (libraries) in the system.

In designing distributed database systems in varied situations, our results would serve as a guideline to estimate a reasonable number of copies of each file to be stored in the systems, provided that the ratio r of the frequencies of update requests to query requests is approximately known in advance.

Type	N	d = 25(%)	d = 50(%)	d = 100(%)
I	30	0.14	0.19	0.21
	50	0.16	0.20	0.22
II	30	0.23(1)	0.23(2)	
	50	0.22(3)	0.21(4)	
III	30	0.19	0.15	0.18
	50	0.14	0.18	0.14

N: the number of sites.
d: the density(%) of links, i.e.,

$$d = \frac{\text{number of pairs (i,j) connected by links}}{\text{number of all pairs}} \times 100.$$

The realized d values in Type II;
(1) 29(%), (2) 55(%), (3) 28(%), (4) 51(%).

Table 1. The values r at which F(r) changes from 2 to 1.

Note: This work was partially supported by A Grant in Aid for Science Research of the Ministry of Education, Science and Culture of Japan.

Acknowledgments

The authors wish to thank Prof. D. Erlenkotter of University of California at Los Angeles for providing them with his code of the simple plant location problem. Thanks are due to Mr. H. Nagase (the president) and Mr. T. Hasegawa of Japan Travel Bureau for the data of file accessing of their computer system. They also wish to thank Prof. T. Kameda of Simon Fraser University for introducing them to the subject of distributed database systems.

References

[1] L.W. Dowdy and D.V. Foster, "Comparative Models of File Assignment Problem", *ACM Computing Serveys*, Vol.14, No.2, pp.287-313, 1982.
[2] D. Erlenkotter, "A Dual-based Procedure for Uncapacitated Facility Location", *Operations Research*, Vol.26, No.6, pp.992-1009, 1978.
[3] S. Muro *et al.*, "Evaluation of the File Redundancy in Distributed Database Systems", *Working Paper*, Dept. of Appl. Math. & Physics, Kyoto Univ., 1983.