

TIME AND COST TRADEOFF FOR DISTRIBUTED DATA PROCESSING

HEESEOK LEE

Department of Management Information Systems, College of Business and Public Administration,
University of Arizona, Tucson, AZ 85721, U.S.A.

(Received for publication 19 January 1989)

Abstract—An important design issue in distributed data processing systems is to determine optimal data distribution. The problem requires a tradeoff between time and cost. For instance, quick response time conflicts with low cost. The paper addresses the data distribution problem in this conflicting environment. A formulation of the problem as a non-linear program is developed. An algorithm employing a simple search procedure is presented, which gives an optimal data distribution. An example is solved to illustrate the method.

INTRODUCTION

Recent advances in communication technologies coupled with the application explosion have led to effective distributed data processing. A distributed data processing system is basically an interconnection of geographically dispersed processing nodes communicating with other nodes through a network. In comparison to centralized data processing systems, distributed data processing systems have certain advantages depending upon the manner in which data are distributed. The distribution of data can be fully characterized by two problems: designing the fragmentation and allocation of fragments [1]. Fragmentation design has rarely been studied. Some methodological approaches are presented in [2, 3]. On the other hand, the allocation of fragments has been widely analyzed in the context of the file allocation problem, since Chu [4] originally investigated the file allocation on a multi-processor system. In [5-8], the objective of the problem is to allocate copies of data files to processing nodes so that the sum of data storage cost and communication cost is minimized. Queueing models have focused mainly on minimal response time as a criterion of performance [9-11]. A comprehensive review of the file allocation problem can be found in [12]. Typically, one of two optimization objectives has been adopted in the design of distributed data processing systems: either minimization of overall operating cost or optimization of some performance-related measurements.

It is noted that the optimization objectives considered separately, may suggest opposed strategies of data distribution. For instance, we can reduce operating cost to put an entire information system at one node where it is cheaper for the system to access data and operate upon it. However, concentrating a large amount of data at one node may mean that a corresponding large number of requests will be transmitted to that node resulting in increased system response time. The objective of this paper is to determine optimal data distribution in view of the tradeoff between system response time and operating cost.

SYSTEM ANALYSIS AND MODEL FORMULATION

An information system in distributed data processing systems is almost invariably relational [13] due to its simplicity of fragmentation. In this case, the information system is a common, global relation itself. The relation is just a mathematical term for a table consisting of tuples and attributes. The information system will be strictly decomposed into non-overlapping fragments. The set of horizontal fragments consisting of subsets of the tuples of the global relation is a good candidate for such a decomposition. Such fragments are appropriate units of allocation and can be referred to as data files. These data files are allocated to N processing nodes. User nodes are assigned to nearest processing nodes to minimize data communication costs. A file request generally involves

a single data file. In this setting, decisions regarding the design of the fragmentation of the information system and allocation of these fragments can be effectively merged together. This decision is referred to as data distribution. In this paper, we deal with this system.

The system operates as follows. The requests are transmitted to the processing node which has the data file containing the data items requested. The requests then queue, receive file service, and finally exit the system.

The following notation is used to present the system:

- I = the index set of processing nodes;
- s_i = the data storage cost at node i per unit time;
- q_{ij} = the communication cost of transmitting query request from node i to node j and transmitting the reply from node j back to node i ;
- u_{ij} = the communication cost of transmitting update from node i to node j ;
- α_i = the mean query request rate at node i ;
- β_i = the mean update request rate at node i ;
- $1/\mu_i$ = the mean service time required to process the message transmitted to node i .

Let us define the decision variable x_i to describe the data distribution:

x_i = the portion of information system stored at node i for $i = 1, \dots, N$.

Since there is only one divisible information system,

$$\sum_{i=1}^N x_i = 1. \tag{1}$$

It is assumed that file requests are generated on a uniform basis. That is, a file request is equally likely to access any of the data items in the system. In this setting, x_i represents the probability that a particular file request will be transmitted to node i . The purpose of the model is to find the optimal x_i . In many instances, it is reasonable to model the file request as a Poisson process. File requests generated in a given user node consist of query requests and update requests. In order to simplify our model, we do not distinguish between the service times for query and update. The system-wide file request rate can be defined by

$$\gamma = \sum_{i=1}^N (\alpha_i + \beta_i). \tag{2}$$

The mean rate of requests transmitted to node i is

$$\gamma_i = \gamma x_i, \tag{3}$$

which also forms a Poisson process.

A performance measure of interest is the system response time of a file request. The system response time can be interpreted as the average time spent in the system by a file request. Denoting this random variable by \bar{T} , we can see that \bar{T} can be obtained as the weighted sum of response time of each node simply using Little's result [14]:

$$\bar{T} = \sum_{i=1}^N \frac{\gamma_i}{\gamma} \bar{T}_i. \tag{4}$$

The \bar{T}_i includes the request processing time, the delay times resulting from the processing of file request, and the computer access time. In most cases, the computer access time is much smaller than the request processing time and can be neglected. A common assumption is that the service requirement of each request is an exponential random variable with mean $1/\mu_i$. Combined with the fact that the arrival process concerning γ_i in (3) is a Poisson process, each processing node can be modeled as $M/M/1$ queue [15] in an open central server network so that

$$\bar{T}_i = \frac{1}{\mu_i - \gamma x_i}. \tag{5}$$

To prevent the queue lengths from growing without bound, it is required that

$$x_i < \frac{\mu_i}{\gamma}, \quad \forall i = 1, \dots, N \quad (6)$$

that is, the mean rate of requests transmitted to a particular node must be less than the mean file service rate. From (4) and (5), the system response time is

$$\bar{T} = \sum_{i=1}^N \frac{x_i}{\mu_i - \gamma x_i}. \quad (7)$$

We can express the operating cost in terms of the data distribution x_i . The operating cost consists of storage cost and communication cost. The expected storage cost per unit time is

$$S = \sum_{i=1}^N s_i x_i. \quad (8)$$

Since the file request arriving at a node may result in updating a local or remote data file, the expected communication costs include query communication cost and update communication cost. Let c_i be the expected system-wide communication cost at node i . Then c_i is simply computed as

$$c_i = \frac{1}{\gamma} \sum_{j=1}^N (\alpha_j q_{ji} + \beta_j u_{ji}). \quad (9)$$

Accordingly, the expected communication cost per unit time is

$$Q = \sum_{i=1}^N c_i x_i. \quad (10)$$

Therefore the expected operating cost per unit time is

$$C = S + Q = \sum_{i=1}^N (s_i + c_i) x_i, \quad (11)$$

or denoting $s_i + c_i$ by d_i ,

$$C = \sum_{i=1}^N d_i x_i. \quad (12)$$

Here d_i is the operating cost incurred at node i .

From (7) and (12), it is pointed out that each of the two cost factors such as system response time and operating cost considered alone, may suggest opposed distribution strategies. In our total system cost function, the tradeoff between system response time and operating cost is characterized by a constant κ as equation (13) below. The total system cost to be minimized is

$$TC = C + \kappa \bar{T}, \quad (13)$$

or

$$TC = \sum_{i=1}^N \left[d_i x_i + \frac{\kappa x_i}{\mu_i - \gamma x_i} \right]. \quad (14)$$

The constant κ thus implies the relative importance of the system response time. Therefore κ can be referred to as the relative cost of the system response time. Finally, the formulation of data distribution is reduced to a nonlinear programming problem:

minimize TC

$$\text{subject to (1), (6), and all } x_i \geq 0, \quad \forall i = 1, \dots, N. \quad (15)$$

OPTIMAL DATA DISTRIBUTION

The optimal solution to the problem (15) is obtained as follows. Our analysis is based on the well-known results of Lagrangian duality [16]. Temporarily relaxing the constraint (6) and applying the *Lagrange multiplier* λ , then the Lagrangian problem is

$$L = TC + \lambda \left(1 - \sum_{i=1}^N x_i \right). \tag{16}$$

Taking the partial derivative of L with respect to each x_i and setting the derivative to zero, we obtain

$$x_i = \frac{\mu_i}{\gamma} - \frac{1}{\gamma} \sqrt{\frac{\kappa\mu_i}{\lambda - d_i}}, \quad \forall i = 1, \dots, N. \tag{17}$$

Let us define the auxiliary function $X_i(\lambda)$ as follows:

$$X_i(\lambda) = \frac{\mu_i}{\gamma} - \frac{1}{\gamma} \sqrt{\frac{\kappa\mu_i}{\lambda - d_i}}, \quad \forall i = 1, \dots, N. \tag{18}$$

And also define

$$X(\lambda) = \sum_{i=1}^N X_i(\lambda). \tag{19}$$

The $X(\lambda)$ is monotonically increasing in λ , where

$$\lambda > \max_{i \in I} d_i. \tag{20}$$

The monotonicity of $X(\lambda)$ implies that we can easily calculate λ numerically so that $X_i(\lambda) = 1$, thus implying that the constraint (1) is satisfied. In this case, a search procedure like the bisection method [17] can be used. The corresponding values of x_i are then obtained from (17).

The constraint (6) is always true of any values of x_i in (17). Hence the only way the set of x_i is not optimal is if one or more of x_i is negative. In this case, we continue the process of setting x_i for the processing node i with the highest π_i to zero, where

$$\pi_i = d_i + \frac{\kappa}{\mu_i}, \quad \forall i = 1, \dots, N. \tag{21}$$

Let us define π_i by allocation cost at a particular processing node i . Here the allocation cost implies costs incurred at the time when data begin to be allocated to a particular processing node. We see that the allocation cost is the sum of d_i and κ/μ_i which indicates the cost factor associated with system response time.

We delete the node set to zero when recalculating new value of λ and the corresponding set of x_i until all the remaining x_i are nonnegative. In summary, we can propose an algorithm for the optimal data distribution x_i^* as follows.

Algorithm 1

- Step 1.* Arrange I such that $\pi_1 \leq \dots \leq \pi_N$.
- Step 2.* Set $H(k) = \sum_{i=1}^k X_i(\pi_k)$, $\forall k = 1, \dots, N$. And set $H(N + 1) = (\sum_{i=1}^N \mu_i)/\gamma$.
- Step 3.* Find k^* such that $H(k^*) \leq 1 < H(k^* + 1)$.
- Step 4.* Find λ^* such that $\sum_{i=1}^{k^*} X_i(\lambda^*) = 1$.
- Step 5.* If $i \leq k^*$, $x_i^* = X_i(\lambda^*)$; otherwise, $x_i^* = 0$.

A proof of the optimality of Algorithm 1 is given in the Appendix. We observe that Algorithm 1 allocates data such that the nodes with relatively high π_i are assigned no data. The other nodes are assigned data depending upon their cost structures.

It is worth noting that the optimum *Lagrange multiplier* λ^* corresponds to the shadow price in conventional linear programming problems. Here the shadow price can be interpreted as the price that the system user would be willing to pay for data allocation. We recognize that the shadow price determines whether or not data is allocated to a particular processing node. In other words,

Table 1. Data of example for distributed data processing

Processing node i	1	2	3	4	5
d_i	0.35×10^{-3}	1.18×10^{-3}	1.90×10^{-3}	3.25×10^{-3}	0.36×10^{-3}
$1/\mu_i$	1.176	0.400	0.185	0.417	3.571

d_i = operating cost at node i per unit time (\$/sec)
 $1/\mu_i$ = mean file service time at node i (min)
 γ = total file request rate = 5.0/min
 κ = relative cost of system response time = $\$1.67 \times 10^{-5}/\text{sec}^2$.

Table 2. Optimal data distribution for example

Processing node i	1	2	3	4	5
π_i	1.53×10^{-3}	1.58×10^{-3}	2.09×10^{-3}	3.67×10^{-3}	3.93×10^{-3}
x_i^*	0.060	0.275	0.665	—	—

π_i = allocation cost at node i (\$/sec)
 x_i^* = optimal data distribution
 λ^* = shadow price = $\$3.16 \times 10^{-3}/\text{sec}$
 C^* = optimal operating cost = $\$1.61 \times 10^{-3}/\text{sec}$
 \bar{T}^* = optimal system response time = 40.4 sec
 TC^* = optimal system cost = $\$2.28 \times 10^{-3}/\text{sec} = \$4140/\text{month}$.

it should not be surprising to discover that if the allocation cost of a particular processing node exceeds its shadow price, then the data should not be allocated to that node.

EXAMPLE

Let us consider a specific distributed data processing system consisting of five processing nodes. Table 1 gives the detailed data for the system. The average rate of requests entering the system is 5 per min. The relative cost of the system response time is assessed to be $\$1.67 \times 10^{-5}/\text{sec}^2$. That is, the importance of 1 min in terms of system response time corresponds to \$1800/month of operating cost.† The example was solved by Algorithm 1 in FORTRAN. Table 2 lists the optimal data distribution.

Some characteristics of the tradeoff between system response time and operating cost are worthy of note. Although the processing node 4 has relatively low file service time, no data is allocated mainly due to its very high operating cost. No data is allocated to the processing node 5 since the benefit from its low operating cost is offset by very high file service time. It is further noted that no data is allocated to the processing nodes of which the allocation costs exceed the shadow price. The overall system cost under the optimal data distribution is \$4140 per month. If data is distributed in the manner in which the system response time is minimized, the optimal system response time can be reduced by 9 sec.‡ However, the optimal system cost under such a single objective is increased to \$4580 per month. The system cost decrease due to the compromise between the two objectives is 9.6%.

CONCLUSION

In this paper, the issue of tradeoff between system response time and operating cost in distributed data processing was explicitly recognized. A nonlinear programming formulation of the problem was presented. The optimal data distribution can be determined by a simple algorithm. The criteria of data distribution was identified. This methodology provides a more realistic instrument for the design of distributed data processing system than the single objective models widely used in the literature.§

†The calculation is based on computer operation at 120 hr per week and 4.2 weeks per month.

‡In this case the objective function is a special case of (14) such that $d_i = 0$ and $\kappa = 1$. The optimum can be obtained by Algorithm 1. In step 4 and 5, x_i^* is obtained analytically without the search procedure:

$$x_i^* = \frac{\mu_i}{\gamma} \frac{\sum_{j=1}^{k^*} \mu_j - \gamma \sqrt{\mu_i}}{\sum_{j=1}^{k^*} \sqrt{\mu_j}}, \quad \forall i = 1, \dots, k^*.$$

§The author would like to express his thanks for the helpful comments from two referees.

REFERENCES

1. S. Ceri and G. Pelagatti. *Distributed Databases: Principles and Systems*. McGraw-Hill, New York (1984).
2. S-K Chang and W-H Cheng. A methodology for structured database decomposition. *IEEE Trans. Software Engng.* **SE-6**, 205-218 (1980).
3. S. Ceri, B. Pernici and G. Wiederhold. Distributed database design methodologies. *Proc. IEEE* **75**, 533-546 (1987).
4. W. W. Chu. Optimal file allocation in a multiple computer system. *IEEE Trans. Comput.* **C-18**, 885-889 (1969).
5. R. G. Casey. Allocation of copies of a file in an information network. *Proc. AFIPS 1972 Spring Jt. Computer Conf.* **40**, 617-625 (1972).
6. H. C. Morgan and K. D. Levin. Optimal program and data locations in computer networks. *Commun. ACM* **20**, 315-322 (1977).
7. B. W. Wah. File placement on distributed computer systems. *IEEE Computer* **17**, 23-32 (1984).
8. H. Pirkul. An integer programming model for the allocation of databases in a distributed computer system. *Eur. J. Opns. Res.* **26**, 401-411 (1986).
9. P. P. S. Chen. Optimal file allocation in multilevel storage systems. *Proc. AFIPS 1973 Natn. Computer Conf.* **42**, 277-282 (1973).
10. E. G. Coffman Jr., E. Gelenbe and B. Plateau. Optimization of the number of copies in a distributed data base. *IEEE Trans. Software Engng.* **SE-7**, 78-84. (1981).
11. M. M. Srinivasan and K. Kant. The file allocation problem—a queueing network optimization approach. *Computers Opns. Res.* **14**, 349-361 (1987).
12. L. W. Dowdy and D. V. Foster. Comparative models of the file assignment problem. *Computing Surveys* **14**, 287-313 (1982).
13. C. J. Date. *An Introduction to Database System: Vol. 1*. Addison-Wesley, Mass. (1986).
14. J. D. C. Little. A proof of the queueing formula $L = \lambda W$. *Opns Res.* **9**, 383-387 (1961).
15. L. Kleinrock. *Queueing Systems: Computer Applications*. Wiley, New York (1976).
16. R. V. V. Vidal. A simple method to solve simple allocation problems. *IIE Trans.* **19**, 234-237 (1987).
17. J. S. Vandergraft. *Introduction to Numerical Computations*. Academic Press, New York (1983).
18. M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, New York (1979).
19. W. I. Zangwill. *Non-linear Programming: A Unified Approach*. Prentice-Hall, Englewood Cliffs, N.J. (1969).

APPENDIX

Proof of the optimality of Algorithm 1: Since $\sum_{i=1}^{k^*} X_i(\lambda^*) = 1 \geq H(k^*) = \sum_{i=1}^{k^*} X_i(\pi_i)$, $\lambda^* \geq \pi_{k^*} \geq \pi_i$, $\forall i = 1, \dots, k^*$, thus implying $0 \leq x_i^* < \mu_i/\gamma$, $\forall i = 1, \dots, N$. Therefore the Kuhn-Tucker conditions [18] are reduced simply to the equations (17) and (1). Since TC is convex and the constraint set is convex, the Kuhn-Tucker conditions are necessary and sufficient for global optimum [19]. The set of x_i^* satisfies these Kuhn-Tucker conditions.