1986

DISTRIBUTED COMPUTER SYSTEM DESIGN FOR LARGE DECENTRALIZED ORGANIZATIONS

Bezalel Gavish
University of Rochester

Hasan Pirkul
The Ohio State University

DISTRIBUTED COMPUTER SYSTEM DESIGN FOR LARGE DECENTRALIZED ORGANIZATIONS

Bezalel Gavish
Graduate School of Management
University of Rochester

Hasan Pirkul
College of Business
The Ohio State University

ABSTRACT

This paper deals with the issue of designing distributed computer systems for large decentralized organizations. Specific problems addressed within this context include: determining the number of computer installations, choosing the sites that will receive these installations, deciding on the sizes of computers at different sites, configuring databases, allocating databases to computer installations and assigning users to computer installations. A class of decentralized organizations is identified for which decisions regarding the database configuration and the allocation of databases among processors can be effectively merged with decisions regarding the assignment of user nodes to processors. For these organizations, the design problem is formulated as an integer programming model. A brief outline of an effective solution procedure is provided and potential uses of the model as a design tool is demonstrated through a number of experiments.

INTRODUCTION

Decentralized organizations generally operate over large geographical regions. Since decision making rights and the demand for decision aids are distributed throughout the operating region, information systems have to provide support at many points dispersed over a large geographic area. Traditional centralized computer systems permit collection and retrieval of data throughout the operating region but concentrate the processing and storage of data in a central site. Such systems are becoming less and less desirable for providing service to decentralized organizations. This is mainly due to deteriorating service levels resulting from increased loads on computer systems. These increased loads are partially caused by increases in size and complexity of organizations. More importantly, there has been a significant change in the nature and scope of information systems. During the last decade, computing and communication costs have declined dramatically. Communication networks having high levels of reliability and availability have become operational. These cost reductions and advances in the information systems technology created new opportunities and changed the way information systems function within organizations. Developments like computer integrated manufacturing, storage and distribution of goods, and point of sale systems reduced the cost of capturing and processing vast amounts of data. Using this data, systems provide managers with timely and accurate information that was not previously available. In many businesses, the mode of operation has changed, e.g., electronic funds transfer, automatic teller machines, long distance transaction processing in the banking sector, and air-

line, hotel, and car rental reservation systems. The net result of all these changes is that information systems place considerably higher loads on the supporting computing systems. These trends, expected to continue in the foreseeable future, favor distributed computer systems which reduce telecommunication costs by placing storage and processing devices closer to the points where data is collected and retrieved (Gavish, 1985).

Distributed computing systems are considerably more complex than centralized systems. Designers of such systems face a number of challenging problems, most of which are nonexistent in centralized designs. Among them is the configuration problem, which may be summarized as determining the number, capacity and location of processing and storage devices, defining databases, and assigning these databases to processors to minimize overall system costs. In this paper we present a model that can be used as a system configuration tool. The configurations provided by the model could be used as starting points in the detailed design of a distributed computer system. Solution procedures for this model are developed and computational results are presented.

LITERATURE SURVEY

The system configuration problem has only recently received attention (Chen and Akoka, 1980; Gavish, 1985; Gavish and Pirkul, 1986(1)). The model presented in this paper is the first to consider processor selection, processor location, database formation and database assignment problems simultaneously. Previous research has generally been concerned with the problem of assigning copies of predefined files to the nodes of a given network of computers. A comprehensive model for this problem, which considered memory limitations and response time constraints, was first developed by Chu (1969). Whitney (1970), Casey (1972), Urano, *et al.*, (1974), Chandy and Hewes (1976), Grapa and Belford (1977), and Ramamoorthy and Wah (1983) investigated the same problem ignoring memory restrictions and response time constraints. Morgan and Levin (1977) and Fisher and Hochbaum (1980) studied a variation of this model in which the allocation of program files and data files were treated separately from each other. Mahmoud and Riordan (1976) presented a model which determined locations of file copies as well as link capacities in a network with pre-determined topology and routing. Coffman *et al.*, (1981) developed a model for determining the optimal number of database copies in a distributed computer system. Lanning and Leonard (1983) presented an algorithm which determines locations of file copies in a store-and-forward computer communications network to minimize the sum of file storage and message transmission costs. Recently, Pirkul (1986) developed a model and presented solution algorithms for the problem of determining contents of databases and allocating these databases among an existing set of processors.

The model presented in this paper differs from the previous research in that it simultaneously considers selecting processors among multiple processor types, determining locations of processing sites, defining databases and assigning these databases to processing sites subject to processor capacities. Most of the previous research assumes that files are predefined, processors are already located and their capacities are given.

PROBLEM DEFINITION, ASSUMPTIONS AND MODEL FORMULATION

Designers of distributed computing systems are faced with the following set of decisions while determining the system configuration:

- How many processing sites should there be in the system,
- In which sites should computers be placed,

- What should be the configurations of computers placed in different sites,

- Which groups of data items should be put together to form database segments,

- How many copies of each database segment to have in the system,

- Where should different database segments be stored,

- What should be the topology of the data communication network.

The objective of the design process is to resolve these issues so that the total cost of implementing and operating the system is minimized. Costs can be separated into two sets. The first set consists of all costs that are incurred in setting up and operating processing sites. Discounted capital costs, rents, payroll, maintenance and utility costs are associated with a given processing site and are defined as a function of the size of the facility and the specific computer configuration placed in it. The second set of costs are the telecommunication costs involved in transferring data between the sites where data is captured and retrieved and the processors where they are stored and processed. The problem is a very difficult one to solve in its general form. In this paper we concentrate our efforts on solving the problem for a class of organizations exhibiting certain characteristics that create significant savings potential in telecommunication costs. These same characteristics also simplify the model and the design process. Furthermore, in many cases, information systems represent an area of great strategic importance for organizations exhibiting these characteristics.

We consider systems that exhibit strong "locality of reference" and that are driven by transaction processing activities. Strong "locality of reference" implies that a data item is generally used in or around the geographic region where it is captured. Activity requiring the data item outside of this region is generally very limited. The main activity in a transaction processing system has to do with adding, deleting and modifying data items associated with various entities. The interpretation of an entity may vary from one system to another but nevertheless, most transaction processing systems can be studied as systems keeping track of various entities. In banking an entity represents a customer account, while in a retail chain it might represent an inventory item. Another characteristic of transaction processing systems is that a transaction generally operates on a single entity or can easily be represented as a series of activities on single entities without leading to inefficiencies, e.g., in the banking sector, a transaction requiring withdrawal from one account and deposit to another can easily be viewed as two different transactions. Firms in the services sector generally exhibit these characteristics. Service organizations such as, insurance companies and credit card firms, are critically dependent on their information systems. Cash, *et al.,* (1983) classify the role of information systems in these organizations as "strategic" and go on to say that the survival of these firms depends on the smooth and efficient functioning of their information systems. Since these organizations also exhibit strong "locality of reference," they are prime candidates for implementing distributed computer systems. In this paper we study the configuration problem for organizations exhibiting the characteristics outlined here.

In systems exhibiting strong "locality of reference" database formation according to geographic proximity represents an effective approach. Each user account is assigned to a single user node. A database partition includes all data items associated with the user accounts belonging to a user node. These partitions are then assigned to processors to form databases. This strategy does not consider data duplication. Since in systems with strong "locality of reference" data items are seldom used outside of a given geographic region, telecommunication cost savings that might result from duplication of data items at different locations are insignificant. For these systems, a strict partitioning of the database without data duplication is a viable strategy. It is desirable to assign data items to processors which are nearest to the geographic regions where these data items are used. It is assumed that each user node is assigned to a host processor and that the database partition containing data items of the accounts associated with that user node is stored in the host processor.

Data communication is accomplished via dedicated links within a metropolitan area and via a

value added network between metropolitan areas. Unless the amounts of data transferred over long distances are very high, an economic incentive exists for using the services of a value added network like TYMNET (1980) and TRANSPAC (Guilbert, 1979).

The system operates as follows. A transaction originating in a user node is directed to the processor which has the database partition containing the records of the account that generated the transaction. This can be easily accomplished as an account number uniquely identifies the customer and the user node to which this account belongs. If the appropriate database partition is assigned to a processor within the same metropolitan area, then the transaction is routed to the processor through a leased local link and the value added network is not used. If the user node and the processor are not in the same metropolitan area, then an intelligent communications controller at the user node directs the transaction to the value added network which transfers it to the appropriate processor.

The following notation will be used to present the model:

I = the index set of user nodes,

J = the index set of processor locations,

K = the index set of available processor types,

S_k = the set of all user nodes and processor sites which are located in the same metropolitan area with processor site (user node) k,

P_k = processing capacity of the processor type k (operations/time period),

V_k = auxiliary storage capacity of the processor type k (bytes),

T_k = telecommunication capacity of the processor type k (bytes/time period),

p_i = processing requirements of transactions directed to the database partition associated with user node i (operations/time period),

v_i = storage requirements for the database partition associated with user node i (bytes),

t'_{ij} = number of transactions per time period that originate at user node i and are routed to the processor which has the database partition associated with user node j (transactions/time period),

r_{ij} = communication load per transaction that originates in user node i and is directed to the processor which contains the database partition associated with user node j (bytes/transaction),

c_{ij} = communication cost via the value added network between nodes i and j ($/byte) (it equals 0 if jeS),

c'_{ij} = rental cost of a local line between user node i and processor site j ($/time period),

c''_i = rental cost of a local line from user node i to the nearest port of the value added network plus connection cost into the value added network ($/time period),

s_{jk} = cost of locating a processing facility of type k in site j ($/time period),

x_{ij} = 1 if database partition associated with user node i is assigned to a processor at site j, 0 otherwise,

y_{jk} = 1 if processor type k is located in site j, 0 otherwise.

Tariffs of value added network companies (e.g., TYMNET) consist of a fixed access cost and a variable cost which is proportional to the amount of data transferred over the network. c''_{ij} captures the local access cost into the value added network, this cost contains the cost of a local line to the nearest value added network port plus the cost of the port itself. Since within a metropolitan area local lines are used instead of the value added network, not every user node incurs port costs. User nodes with database segments assigned to a processing site within the same metropolitan area incur only line rental costs (c'_{ij}). The communication costs which are incurred when data is transmitted over the value added network is captured by c_{ij}. If we let,

$c_{ij} = c_{ij}'$ if user node i and processor site j are in the same metropolitan area,

c''_i if user node i and processor site j are not in the same metropolitan area,

then the model can be stated as:

Problem-IP

$$Z_{IP1} = MIN\{\Sigma_{i\in I}\Sigma_{j\in J}(c_{ij}+\Sigma_{l\in I} f_{li} r_{li} c_{lj}) x_{ij}\}+\Sigma_{j\in J}\Sigma_{k\in K} s_{jk} y_{jk} \quad (1)$$

Subject to:

$$\Sigma_{j\in J} x_{ij}=1 \qquad \forall\, i \#\in\# I \quad (2)$$

$$x_{ij}-\Sigma_{k\in K} y_{jk}\leq 0 \qquad \forall\, j \#\in\# J,\ i \#\in\# I \quad (3)$$

$$\Sigma_{k\in K} y_{jk}\leq 1 \qquad \forall\, j \#\in\# J \quad (4)$$

$$\Sigma_{i\in I} p_i x_{ij} \leq \Sigma_{k\in K} y_{jk} p_k \qquad \forall\, j \#\in\# J \quad (5)$$

$$\Sigma_{i\in I} v_i x_{ij} \leq \Sigma_{k\in K} y_{jk} v_k \qquad \forall\, j \#\in\# J \quad (6)$$

$$\Sigma_{i\in I} t_i x_{ij} \leq \Sigma_{k\in K} y_{jk} t_k \qquad \forall\, j \#\in\# J \quad (7)$$

$$x_{ij}\in\{0,1\}\forall i\in I, j\in J; y_{jk}\in\{0,1\}\forall j\in J, k\in K \quad (8)$$

Where t_i is defined as,

$$t_i = \Sigma_{l\in I} f_{ji} r_{li}$$

The first component of the objective function represents telecommunication costs. The second component captures costs associated with establishing processing sites. Constraint set (2) states that a database partition can only be assigned to one processor. Constraint set (3) ensures that there is a processor at every site to which at least one database partition is assigned. Constraint set (4) prohibits more than one processor being assigned to the same site. Constraint sets (5), (6), and (7) reflect the capacity constraints on the processing, storage and communication capabilities of the processors. Constraint set (8) represents integrality conditions on variables.

SOLUTION PROCEDURE

Problem-IP is an NP-Complete problem (Gavish and Pirkul, 1986(b)). Commercial integer programming packages can solve small instances of this model but can not handle realistic size problems. This point becomes clearer by considering that a problem with 100 user nodes, 20 potential processor sites and 5 processor types has 2100 integer variables and 2180 constraints. Due to the size and complexity of problem-IP, it is unrealistic to expect that optimal solutions can be found by exerting realistic levels of computational effort. Therefore, we have concentrated on the development of an efficient heuristic solution procedure that takes advantage of the problem structure. This procedure was developed using a Lagrangian relaxation approach. In this approach, constraint set (2) is dualized and the remaining problem is decomposed over processor location variables. The resulting subproblems are then solved as a series of multiconstraint knapsack problems (Gavish and Pirkul, 1985(a); 1985(b); 1985(c)). The solution of the Lagrangian relaxation not only provides a lower bound on the optimal solution value of the original problem but it is also used to obtain a feasible solution for that problem.

In the solution to the relaxed problem only constraints belonging to set (2) may be violated. This implies that there may be user nodes which are not assigned to processors or there may be user nodes which are assigned to multiple processors. If these violations are corrected the resulting solution will be a feasible solution for the original problem. Since there may be a number of unassigned user nodes, the processing capacity provided by the solution of the relaxed problem may not be sufficient to serve all the user nodes. If this is the case, computer type k at site j that was not picked in the solution of the relaxed problem and would result in the least increase in the objective function of that problem is designated open and the solution of the corresponding subproblem is used to augment the solution of the relaxed problem. This

process is repeated until sufficient capacity is attained. At that point all but the least costly assignment of those user nodes that are assigned to multiple processors are dropped. Next, those nodes that were unassigned are assigned to processors that are least costly to access, generating a feasible solution. If a particular assignment will cause a violation of one of the capacity constraints, the next least costly assignment is considered. Technical details of this procedure are reported by Gavish and Pirkul (1986(b).

The procedure was coded in FORTRAN IV and a number of computational experiments were performed to test the quality of solutions it provides. A total of 320 problems were solved during these experiments. Problem sizes range from 60 to 100 demand points and 10 to 20 processor sites. Five different processor sizes are considered. Problems were arranged into sets of 10. All problems in a set were generated to have the same structure using the following method. The coordinates of both demand points and processor sites were drawn from a uniform distribution over a rectangle with sides 50 and 100. The euclidian distance between demand point i and processor site j is then rounded to the nearest integer value (d_{ij}) which is used to define cost coefficients c_{ij} in the following manner:

$$c_{ij} = d_{ij} * W1$$

where c_{ij} represents c_{ij} + S t' r c and W1 is a positive constant. Processor location costs are defined as:

$$s_{jk} = [1+(k-1)*.2] * (W2 + W2 * R/4)$$

where W2 is a positive constant and R is a random number drawn from a uniform distribution between 0 and 1. It should be noted that W1 and W2 can be varied to create problems with different levels of trade-off between processor location and communication costs. The resource consumption of each demand point is determined using the following formulas:

$$p_i = 50 * R$$

$$s_i = (3/4) * p_i + (p_i/2) * R$$

$$t_i = (3/4) * p_i + (p_i/2) * R$$

where R is a random number drawn from a uniform distribution between 0 and 1. Consumption of resources are generated to be correlated with each other as one would expect this to be the case in real life. Processing, communications and storage capacities of processors are defined as:

$$P_k = [1+(k-1)*.2] * 500$$

$$S_k = [1+(k-1)*.2] * 500$$

$$T_k = [1+(k-1)*.2] * 500$$

With these parameters, the smallest processor will be able to handle 20 demand points on the average and the largest one will handle around 40 demand points.

Results of computational experiments are reported in Table 1. Gaps between feasible solution values and lower bounds expressed as percentages of feasible solution values are reported. For each group of problems minimum, maximum and mean gap values are reported. Since the gap between the optimal solution value and the heuristic solution value is always less than the gap between the heuristic solution value and the lower bound, the gaps reported in Table 1 provide us with a good estimate of the solution quality. This is an important advantage of the solution procedure used here. In most problems the gaps are not significant. In fact, in many problems optimal solutions are found as pointed out by the absence of a gap. The gaps are larger in problems where processor location costs totally dominate communication costs resulting in solutions with a minimum number of processors. Also reported are average computing times in CPU seconds of an Amdahl 470 V/8. These times range from 2.6 to 142.6 seconds.

The model can be used to solve larger problems than those reported here. However, computational experiments show that there are significant increases in solution times beyond 200 user nodes and 20 potential service points. Problems with sizes beyond that point can be handled through the model by clustering a number of user points within the same metropolitan area into a single user node. This would effectively reduce the problem size and make it possible to solve the problem with a reasonable level of computational effort. Furthermore, within the context of this model the inaccuracies introduced by such clustering will be negligible.

Table 1. Performance of the Solution Procedure.

Problem Id.				Gap Between Solution Value and Lower Bound (% of Lower Bound)			Solution Time
W2	W1	M	N	Min.	Mean	Max.	CPU Sec.
200	4	10	60	.00	.00	.02	2.6
			80	.01	.21	.71	7.7
			100	.00	.10	.19	9.4
200	4	20	60	.00	.04	.18	6.3
			80	.00	.04	.18	10.2
			100	.00	.06	.16	11.3
200	2	10	60	.00	.34	.83	7.9
			80	.19	.33	.57	22.7
			100	.18	.57	1.48	34.1
200	2	20	60	.00	.73	3.66	15.0
			80	.00	.24	.75	22.5
			100	.00	.41	1.31	34.2
400	4	10	60	.00	.31	.81	8.4
			80	.18	.57	1.40	22.3
			100	.10	.65	1.39	36.8
400	4	20	60	.00	.11	.38	12.1
			80	.00	.19	.42	20.8
			100	.03	.45	1.57	28.8
400	2	10	60	.09	2.55	4.56	28.0
			80	.15	1.22	5.37	45.4
			100	.20	1.03	2.01	98.5
400	2	20	60	.35	2.00	5.59	59.2
			80	1.43	2.76	3.55	93.1
			100	1.23	2.58	3.93	142.6

Uses of the Model in Distributed Computer Systems

This section illustrates the kinds of questions that can be answered through the use of this model. It should be emphasized that our goal is not to reach conclusions that will hold true in every case but rather demonstrate the use of the model to study a number of issues relevant to the design of distributed computer systems.

Effects of Using Multiple Processor Configurations

Demand points in an information system generally form various size clusters. In most networks these clusters represent metropolitan areas. Typically, the size of the demand depends on the size of the town which in turn determines the number of demand points in that town. Due to this clustering effect it is expected that the system will include a number of different processor configurations.

An experiment was designed to test for the effects of the number of available processor configurations on the design reached by the model. First, a group of five problems were solved by making only two processor configurations available. Next, the same problems were solved with 5 different processor configurations. The resource capacities of the smallest configuration were set so that it could handle around 20 demand points on the average. The other configurations were obtained by increasing all three resources proportionally. The largest configuration that was considered could handle around 40 demand points. The results are reported by normalizing the processor capacities such that the smallest processor is defined as having capacity 1 and the largest one as having capacity 2. The problems were generated to have 80 demand points and 10 potential processor sites. The processor location and communication cost components were arranged so that there was a degree of trade-off between these cost components. The cost of different processor configurations was determined by assuming that 30 percent of the cost related to the smallest configuration is independent of the size of the processor. This 30 percent can be attributed to fixed costs such as accommodation costs. The remaining 70 percent was assumed to increase proportionally with the size of the processor.

The results of the experiment are summarized in Table 2. In the first case, there were only two configurations, one of which was twice the size of the other. In the second case, three more configurations were introduced. The sizes of these additional configurations were chosen to be between those of the first two. Table 2 reports the number of processors installed from each configuration as well as the processor location cost, communication cost and total cost for each problem. The costs are reported on a scale such that the average total cost for the first case is set equal to 1. The results confirm our expectations that as the number of processor configurations is increased systems costs will decrease. This is due to a decline in processor costs. In the first case, demand is satisfied by installing the smaller processors, since there is not enough demand concentrated in one location to justify installation of a large processor. Consequently, there are some processors which are clearly underutilized. In the second case, the newly provided medium size processors are used and the number of processors is generally reduced. The processor installations are also found to be better utilized. This leads to a reduction in processor location costs. On the other hand, communication costs are found to increase. This can be explained by the fewer number of processors that are being used. Since the increase in communication costs is lower than the decrease in processor location costs in all cases, the total systems costs are found to decline.

Effects of Reduction in Computing Costs

Computing costs have declined significantly over the last two decades and are expected to do so in the future. Since these costs constitute a significant portion of the processor location costs captured in our model, it will be interesting to see how a reduction in hardware costs will affect the configuration of a distributed computing system provided by the model. An experiment was designed to investigate this issue. Again, a group of five problems with 80 demand points and 10 potential processor sites was solved. Each problem included the same five processor configurations used in the previous experiment. Processor location costs were assumed to have the same structure as before. These costs were reduced by 6 percent in each of the eight time periods while communication costs were kept unchanged.

Table 2. Effects of Having a Number of Different Processor Configurations.

Case 1. A System with Two Different Processor Configurations

Number of Processors Used		Cost		
Processor Size 1	Processor Size 2	Processor Location	Communication	Total
5		.406	.623	1.029
5		.406	.541	.948
3	1	.382	.627	1.008
5		.406	.581	.987
5		.406	.622	1.028

Case 2. A System with Five Different Processor Configurations

Number of Processors Used					Cost		
Processor Size 1	1.25	1.50	1.75	2	Processor Location	Communication	Total
3		1			.353	.651	1.004
2	2				.353	.577	.930
3	1				.339	.654	.993
2	2				.353	.596	.949
1	1	1			.286	.724	1.010

The results of this experiment are reported in Figures 1, 2 and 3. In Figure 1, average processor location, communication and total costs are reported for each time period. These costs are reported on a scale which sets the average total cost for the first period equal to 1. It is observed that the total costs have declined by about 19 percent over eight time periods. If we had kept the original configurations and reduced the processor location costs, this decline would be around 16 percent. The additional 3 percent reduction is due to the impact of trade-off between processing and communication costs. As processing costs are reduced, total processing capacity is both increased and further distributed resulting in a decline in communication costs. This decline accounts for 4 percent of the 19 percent overall decline. The remaining 15 percent is due to reduced processor location costs. This implies that over time total processor capacity is increased around 3 percent. This increase in processing capacity coupled with further distribution of this capacity results in a 6 percent decline in communication costs (this is equivalent to a 4 percent decline in total costs). The increase in total processing capacity is generally attained through additional processing sites. The average size of a processor for each time period is plotted in Figure 2. This graph clearly shows a decline in the average processor size. This decline is a function of the underlying demand points and can be considerably larger for problems generated by using a different structure, such as clustered demand

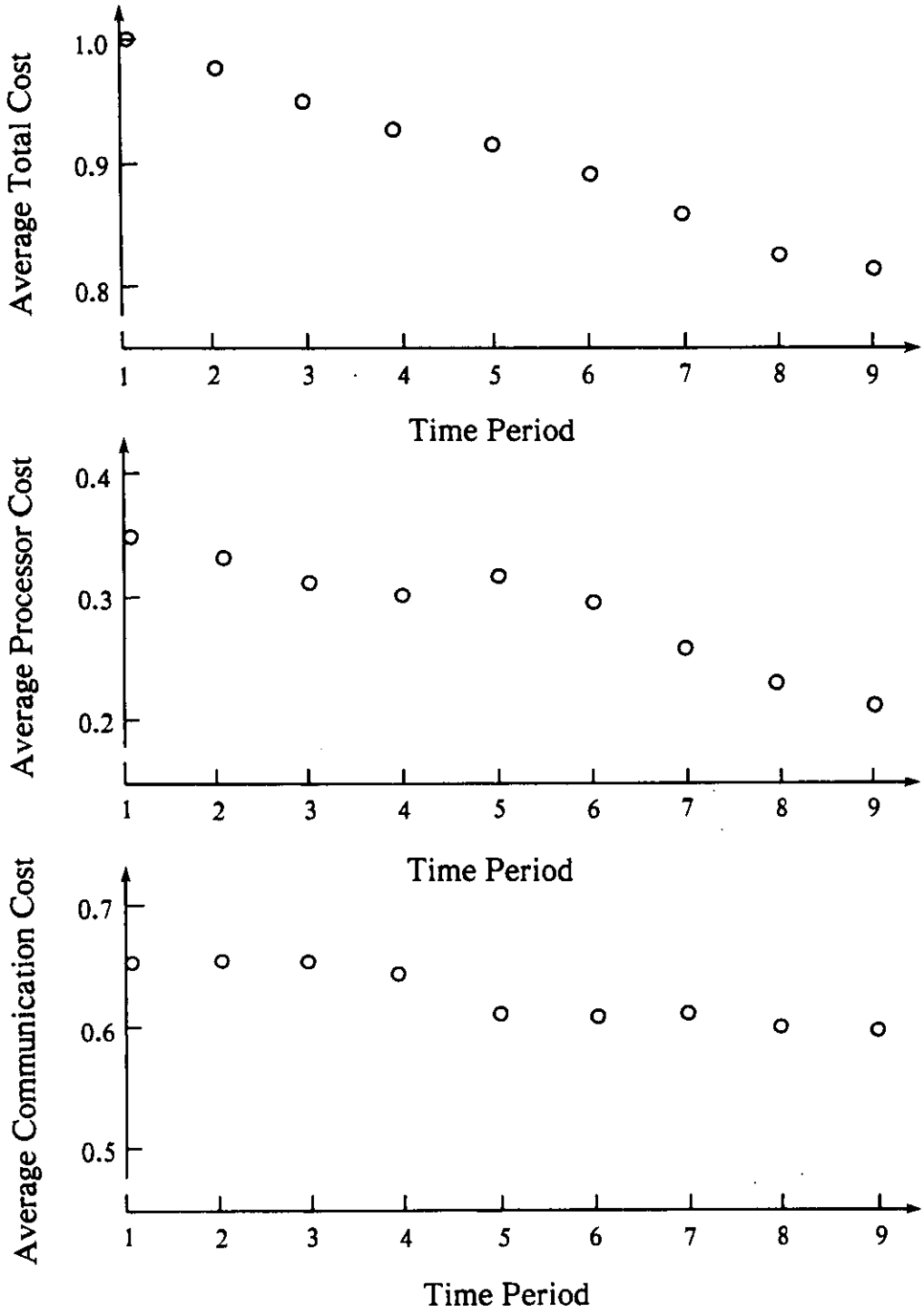


Figure 1. Changes in System Costs as Hardware Costs are Reduced.

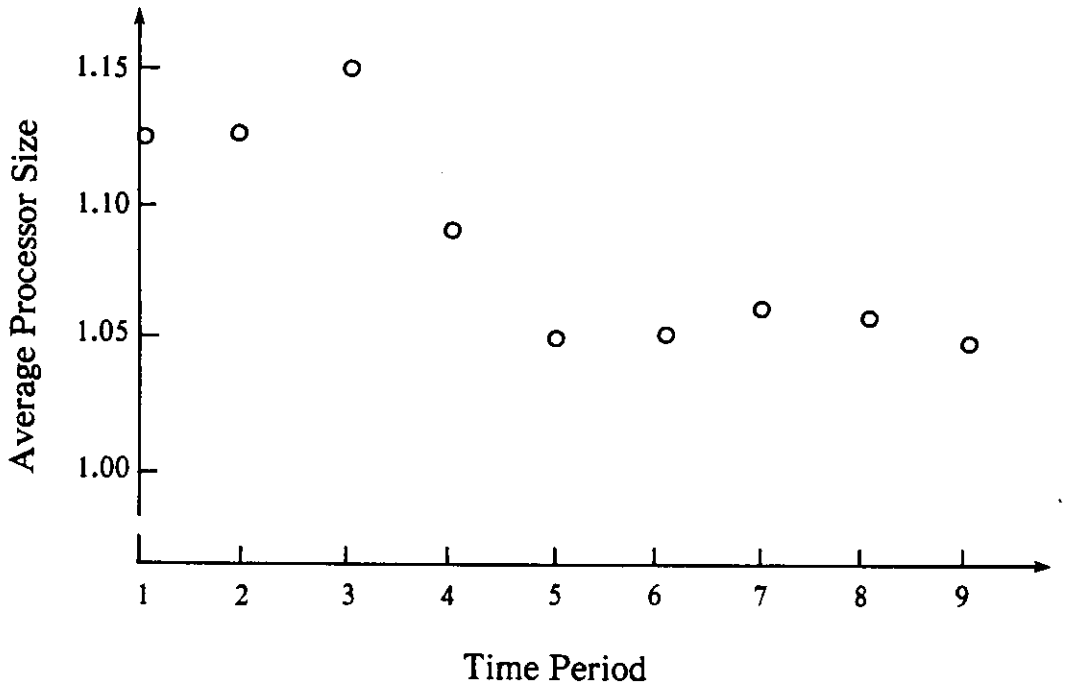


Figure 2. Average Processor Size as Hardware Costs are Reduced.

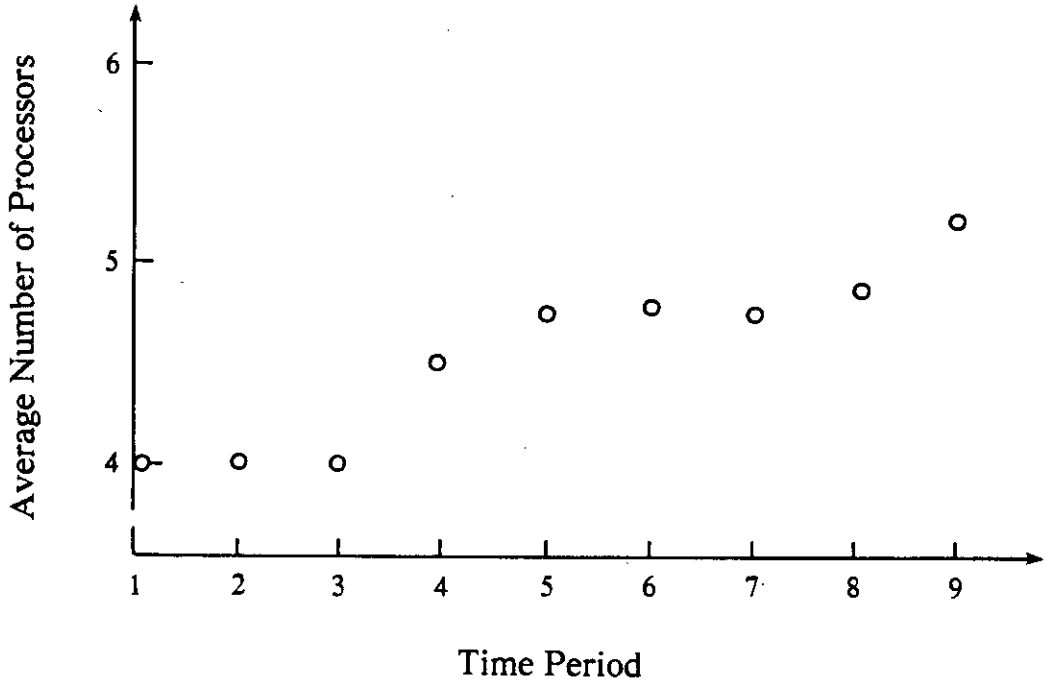


Figure 3. Average Number of Processors.

points. The average number of processor installations is presented in Figure 3, which shows an increase in the number of processors as reported previously. These results suggest that as processing costs decline it is reasonable to expect an increase in the number of processing sites.

Effects of Hardware Pricing Strategies

Up to this point, we have used a processor location cost structure which consisted of a fixed and a variable portion. The variable portion was introduced to capture the hardware costs and was assumed to be proportional to the processor size. This simple structure does not completely capture the pricing strategies observed in the market place. Microcomputers, minicomputers and mainframes are not easily substituted for one another due to the underlying system structure. As a result, hardware vendors treat microcomputer, minicomputer and mainframe markets as different market segments and adjust their pricing strategies accordingly. Therefore, we do not observe a fixed price performance ratio as we move from small processors to the large ones. An experiment was designed to see the impact of different cost performance ratios on the system design. As before, a set of five test problems was used. Each problem consisted of 80 demand points, 10 potential processor sites and 5 processor configurations. This set of problems was first solved keeping the price performance ratio equal for all configurations. The ratio was then decreased for the second configuration and the problems were resolved. Finally, the ratio was decreased for the third configuration and the experiment was repeated. The results of this experiment are reported in Table 3. As expected, the processor location costs and the total costs decline for the second

Table 3. Effects of Pricing Strategies by Hardware Vendors.

	Number of Processors Installed					Cost		
	Processor Size					Processor		
	1	1.25	1.50	1.75	2	Location	Communication	Total
	3		1			.362	.666	1.028
	2	2				.362	.590	.952
Case 1	3	1				.347	.669	1.016
	2	2				.362	.610	.971
	1	1	1			.293	.740	1.033
	1	3				.349	.669	1.018
	2	2				.343	.590	.934
Case 2	2	2				.343	.669	1.013
	1	3				.349	.604	.953
	1	1	1			.284	.740	1.024
	3		1			.349	.666	1.015
	2	2				.362	.590	.952
Case 3	3		1			.349	.648	.997
	2		2			.366	.602	.967
	1	1	1			.280	.740	1.021

and third runs. We can also observe a shift toward configurations which provide a better price performance ratio. The average number of processors used rose from 1.2 to 2.2 for the second configuration and from .4 to 1 for the third configuration as their price performance ratios were lowered.

Effects of Access Patterns

Information generation, flow and access patterns vary widely across organizations. We expect that information systems will reflect these underlying patterns. It was previously mentioned that most service organizations tend to be well served by distributed processing systems. This is mainly due to strong locality of reference in such organizations. In these systems, a data item is generally referenced from the location where it originates. We expect that the ratio of local accesses to long distance accesses plays an important role in determining the configuration of the information system. As this ratio decreases, the savings in communication costs also decline. Since long distance accesses are dominant, the geographical proximity of a service point to a processor becomes less significant.

An experiment was performed to investigate the locality of reference effects on system design. Two sets of problems were generated. The first set consisted of problems with a high ratio of local to long distance accesses. In the second set this ratio was lowered. Each problem consisted of 80 demand points, 10 potential processor sites and 5 different processor configurations. Processor location costs were generated using the same method described in section 5.1. The results of this experiment are summarized in Table 4. As the percentage of local transactions decreases, communication costs increase and overall system costs are higher. On the other hand, since potential communication cost savings decline, processor costs become more significant and the model moves towards decreasing processor costs. This observation is supported by an increase in the average size of a processor and a decrease in the average number of processor installations.

Table 4. Effects of the Degree of "locality of reference."

Number of Processors Used					Cost		
Processor Size					Processor		
1	1.25	1.50	1.75	2	Location	Communication	Total
3		1			.362	.666	1.028
2	2				.362	.590	.952
3	1				.347	.669	1.016
2	2				.362	.610	.971
1	1	1			.293	.740	1.033

Number of Processors Used					Cost		
Processor Size					Processor		
1	1.25	1.50	1.75	2	Location	Communication	Total
	1	2			.321	.778	1.099
4					.332	.730	1.062
	2	1			.307	.785	1.093
2	2				.362	.719	1.081
1	1	1			.293	.783	1.076

Clustered Demand Points

In most service organizations, demand points are clustered around metropolitan areas. Generally, one would expect to serve the whole cluster by a single processor if the cumulative demand of the cluster justifies locating a processor in or near that cluster. If the demand is more than what can be handled by a single processor, a second one might be required. An alternative to this move is to assign the excess demand to processors primarily serving other clusters. Therefore, one would expect to see a strong relationship between the number of clusters and the number of processor installations. An experiment was designed to demonstrate this relationship. Two groups of 5 problems were generated, the first one with 5 clusters and the second one with 3 clusters. Each problem consisted of 100 demand points, 10 potential processor locations and 5 different processor configurations. The problems were similar in structure to the problems used in the previous experiments. The capacities of processors were set so that the smallest one could handle 20 demand points and the largest one could handle around 40 demand points. The problem structures of each group are summarized in Figures 4 and 5. In these figures, the location and number of demand points in each cluster are displayed. Also displayed are the locations of potential processor sites.

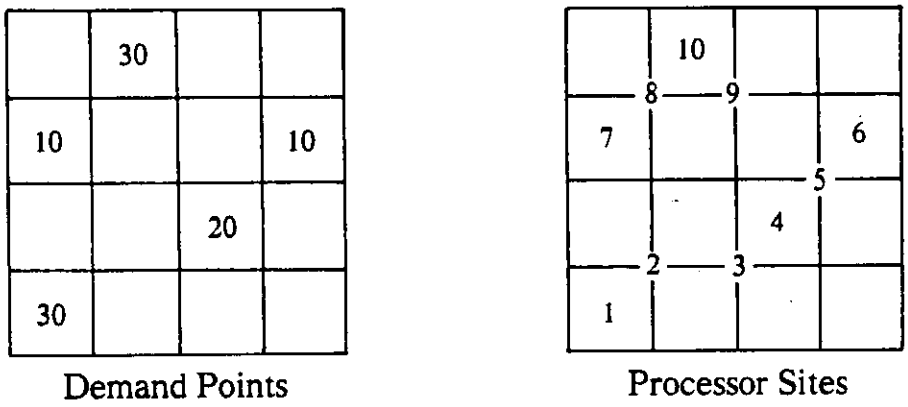


Figure 4. Topology for the Five Cluster Case.

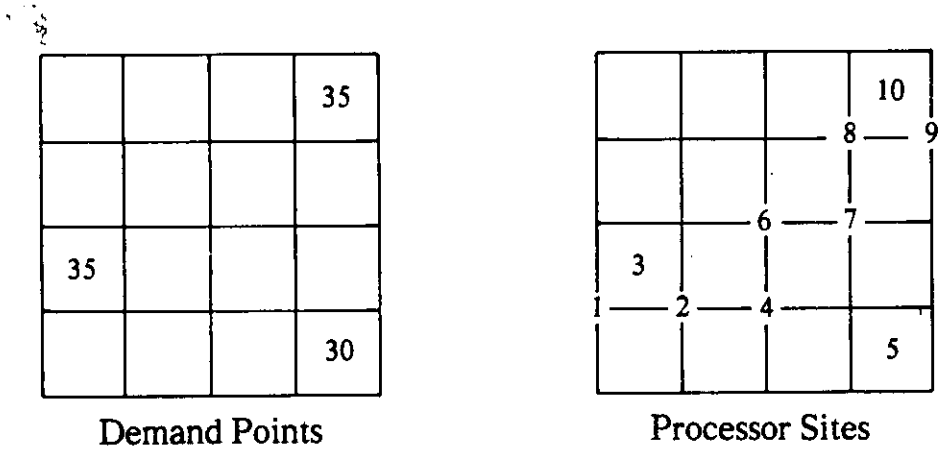


Figure 5. Topology for the Three Cluster Case.

The results of the experiment are reported in Tables 5 and 6 and Figure 6. In these Tables, the sites of processor installations and the size of the processor in each installation are displayed. Also processor location, communication and total costs are reported. Figure 6 illustrates the first problem reported in Table 6 in more detail. From these tables, it can be observed that the number of processor installations is almost always equal to the number of clusters. This is due to the fact that we provided a variety of processor sizes which can match the demands of the clusters generated. Both processor location and communication costs are lower for the case with fewer clusters. This is mainly because processors are better utilized due to the greater size of the clusters in that case.

Changes in System Design Under Increasing Demand

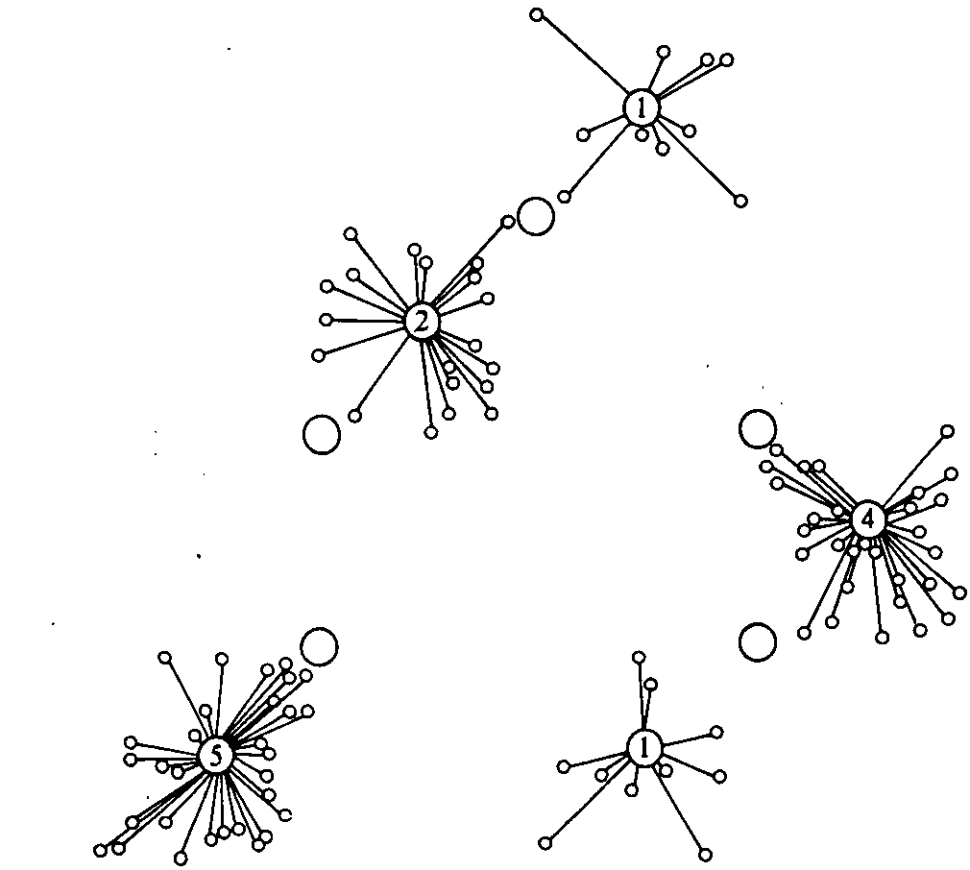
Computer information systems tend to be very dynamic. Services demanded from these systems change rapidly over time, necessitating modification and expansion of the system. Therefore, when a system is designed, the designer is required to make plans for smooth expansion of the system over time. Demand increases can take two different forms. First, due to the growth of the business at the existing facilities, the demand might be increasing relatively uniformly throughout the system. Secondly, the demand might increase as a result of expanding into new areas. These two cases will have dif-

Table 5. Designs for the Five Cluster Case.

Sites at Which Processors Are Located (Processor Size)					Cost		
1	1.25	1.50	1.75	2	Processor Location	Communication	Total
6,7	4	10	1		.446	.577	1.023
4,6,7	10	1			.420	.582	1.002
4,6,8	10	1			.420	.558	.978
4,6,7		10	1		.446	.557	1.003
4,6,7		10	1		.446	.548	.994

Table 6. Designs for the Three Cluster Case.

Sites at Which Processors Are Located (Processor Size)					Cost		
1	1.25	1.50	1.75	2	Processor Location	Communication	Total
1			5	3,10	.450	.522	.972
		3,5,10			.348	.564	.912
		3,5		10	.334	.543	.877
			5	3,10	.374	.503	.877
			5	3,10	.374	.539	.913



(m) Indicates that a processor of type m is located

◯ Potential porcessor site

o Demand points

Figure 6. The Design for the First Problem of the Five Cluster Case.

ferent effects on the system configuration. It is very likely that in a given organization the demand increase will involve a combination of both of these forms. We will treat them separately to distinguish the effects of each one. An experiment was designed with this purpose in mind. In the first part of the experiment, the number of demand points was kept unchanged and the demand for services from each demand point was increased uniformly over eight time periods. The demand was increased by about 10 percent in each period. In this experiment, the problems consisted of 80 demand points, 10 potential processor sites, and 5 different processor configurations. In the second part of the experiment, a demand increase of 20 percent was introduced as a result of a new cluster being added to the original problem. The problems were obtained by appending a new cluster containing 20 demand points to each of the problems used in the initial run of the first case.

The results of this experiment are reported in Figures 7, 8 and 9 and Table 7. In Figure 7 average communication, processor location and total costs are plotted as a function of uniformly increasing demand. In Figures 8 and 9, the

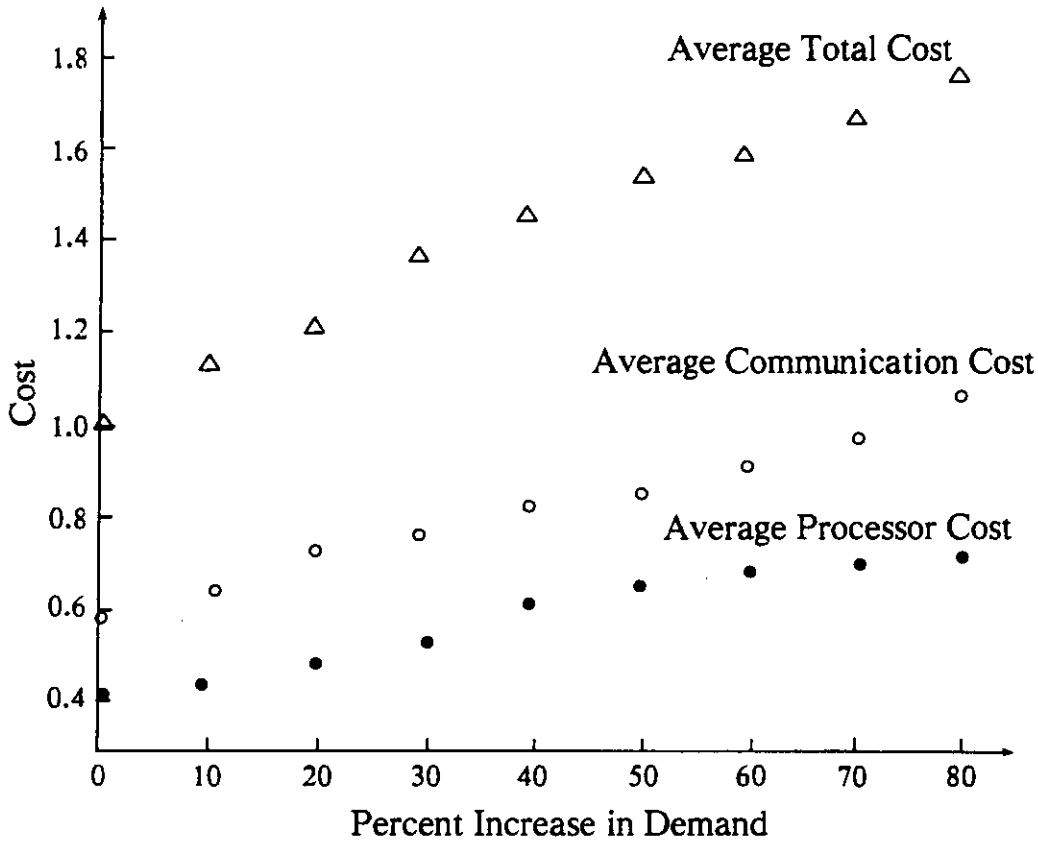


Figure 7. System Costs Under Uniformly Increasing Demand.

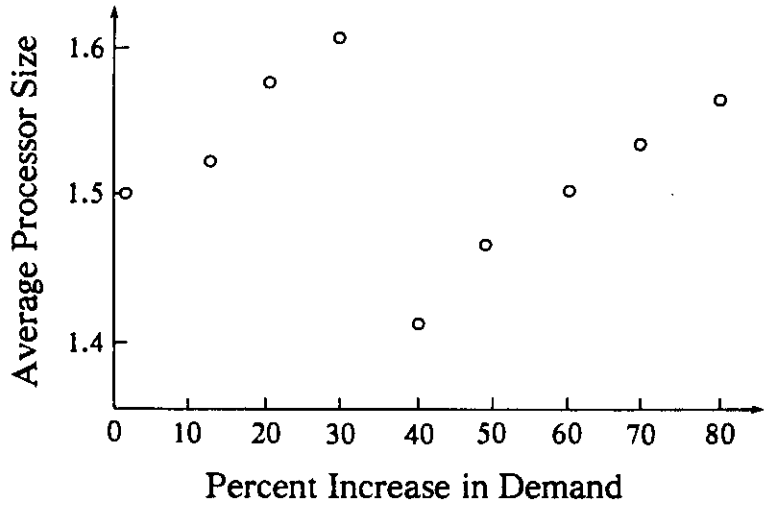


Figure 8. Average Processor Size Under Increasing Demand.

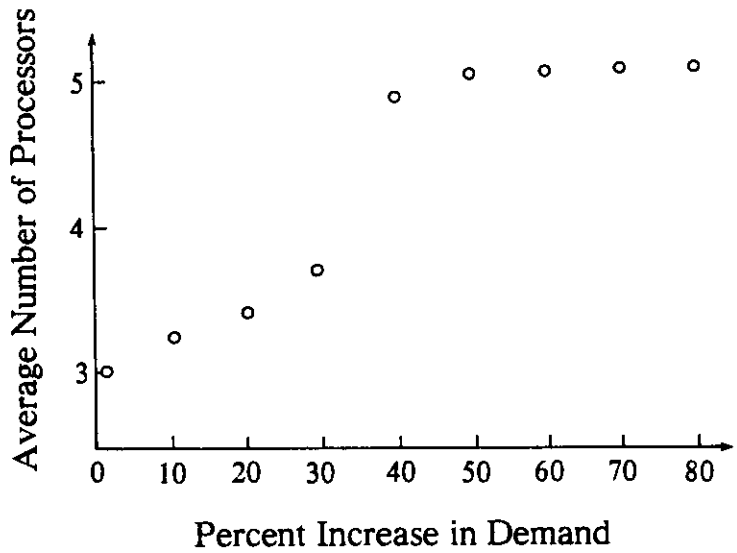


Figure 9. Average Number of Processors Under Increasing Demand.

Table 7. Change in the System Configuration as a New Cluster of Demand Points.

The Results of the Run with the Original Problems

Number of Processors Used					Cost		
Processor Size					Processor		
1	1.25	1.50	1.75	2	Location	Communication	Total
1				2	.408	.582	.990
	1			2	.426	.597	1.022
1			1	1	.390	.602	.993
1				2	.408	.581	.989
	1		1	1	.408	.598	1.066

The Results of the Run with the New Cluster Added

Number of Processors Used					Cost		
Processor Size					Processor		
1	1.25	1.50	1.75	2	Location	Communication	Total
2				2	.509	.730	1.239
1	1			2	.527	.746	1.272
2			1	1	.491	.743	1.234
1			1	2	.544	.720	1.264
	2		1	1	.527	.753	1.279

average processor size and the average number of processor installations are reported. The costs reported in Figure 7 are scaled so that the average total cost for the initial period is set equal to 1. It is observed that the processor location, communication and total costs increase nearly uniformly over time. These increases are found to be proportional to the increase in demand. For an 80 percent increase in demand, total cost increases by about 77 percent, communication cost increases by around 80 percent and the processor cost increases by around 72 percent. These figures all point to the fact that uniform demand increases do not change the structure of the problem. The average number of processors increases up to a point and then stays relatively stable. Initially, the increasing demand results in opening new processor installations. Once the number of processor installations match the number of demand point clusters, the increase in demand is satisfied by increasing the processor size at the existing installations. This is clearly shown in Figures 8 and 9. The average processor size drops at the point where the number of processors jump to near five. Then the average number of processors stay constant while the average processor size steadily increases.

When demand increases take the form of business expansion into new areas, the underlying structure of the problem changes. In the second part of the experiment, a new cluster of demand points is introduced and the change in the system design is observed. The results reported in Table 7 clearly indicate that, depending on the size of the new cluster, the demand increase is likely to lead to the opening of a processor installation around the new cluster. In all but one of the five problems solved, the original set of processors is retained and a new processor is opened around the new cluster.

It is possible to conclude from this experiment that beyond a given demand level each cluster will have a processor to serve the points within that cluster. More importantly, this experiment shows that we should not be expanding processing capacity strictly by opening new sites or by expanding the existing sites. The optimal policy is a combination of both of these approaches. Therefore, systems should be designed to have the flexibility of expanding capacity in the existing processing sites as well as to open new processing sites over time.

SUMMARY AND CONCLUSIONS

The problem of designing computer systems for large decentralized organizations is studied. The problem is modelled as a mathematical program. Various potential uses of the model are demonstrated through a number of experiments. This model can be used as a design tool by providing preliminary designs which can serve as good starting points in the overall design of a distributed computer systems for organizations exhibiting strong "locality of reference." Since there are many aspects of the problem that are not captured by this model, in most cases the designs generated by the model can not be directly implemented. The designer can experiment with the model and reach a better understanding of the various design issues. He can then consider the information provided by the model along with other relevant information, such as management style of the firm, to arrive at the final design. The model is used in a number of experiments to study various issues effecting the system configuration. As a result of these experiments the following conclusions are reached:

- Using a family of compatible processors with different capacities leads to more cost effective systems then standardizing on a single processor type.

- As processing costs decline relative to communication costs, processing capacity and number of processing sites should increase leading to savings in communication costs.

- As the "locality of reference" level increases, processing power is distributed more and more

- Number of processing sites are strongly related to the number of demand point clusters and sizes of these clusters. Beyond a certain combined demand level it is best to locate a processing site within a cluster rather than serve the demand points in that cluster from a processor located somewhere else.

- Capacity expansion should not be

attained only by opening new processing sites or by expanding the existing ones. The best policy is one which combines both of these strategies.

REFERENCES

Casey, R.G., "Allocation of Copies of a File in an Information Network," *AFIPS Conference Proceedings,* Volume 40, 1972, pp. 617-625.

Cash, Jr., J. J., McFarlan, F. W. and McKenney, J. L. *Corporate Information Systems Management: Text and Cases,* Richard D. Irwin, Inc., Homewood, Illinois, 1983.

Chandy, K. M. and Hewes, J. E. "File Allocation in Distributed Systems," *International Symposium on Computer Performance Modeling Measurement and Evaluation,* 1976, pp. 10-13.

Chen, P. and Akoka, J. "Optimal Design of Distributed Information Systems," *IEEE Transactions on Computers,* Volume C-29, Number 12, 1980, pp. 1068-1080.

Chu, W. W. "Optimal File Allocation In Multiple Computer Systems," *IEEE Transactions on Computers,* Volume C-18, 1969, pp. 885-889.

Coffman, Jr., E. G., Gelenbe, E. and Plateau, B. "Optimization of the Number of Copies in a Distributed Data Base," *IEEE Transactions on Software Engineering,* Volume SE-7, Number 1, 1981, pp. 78-84.

Fisher, M. L. and Hochbaum, D. S. "Database Location in Computer Networks," *Communications of the ACM,* Volume 27, Number 4, 1980, pp. 718-735.

Gavish, B. "Models for Configuring Large Scale Distributed Computing Systems," *AT&T Technical Journal,* Volume 64, Number 2, 1985, pp. 491-532.

Gavish, B. and Pirkul, H. "Zero-One Integer Programs with Few Constraints - Lower Bounding Theory," *European Journal of Operational Research,* Volume 21, 1985, pp. 213-224.

Gavish, B. and Pirkul, H. "Zero-One Integer Programs with Few Constraints - Efficient Branch and Bound Algorithms," *European Journal of Operational Research,* Volume 22, 1985, pp. 35-43.

Gavish, B. and Pirkul, H. "Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality," *Mathematical Programming,* Volume 31, 1985, pp. 78-105.

Gavish, B. and Pirkul, H. "Computer and Database Location in Distributed Computer Systems," *IEEE Transactions on Computing,* Volume C-35, Number 7, 1986, pp. 583-590.

Gavish, B. and Pirkul, H. "An Integer Programming Model and Solution Procedures for Designing Distributed Computer Systems," Working Paper, College of Business, The Ohio State University, Columbus, Ohio 1986.

Grapa, E. and Belford, G. G. "Some Theorems to Aid in Solving the File Allocation Problem," *Communications of the ACM,* Volume 20, Number 11, 1977, pp. 878-882.

Guilbert, J. F. "TRANSPAC: The Impact of a Value Added Network," *Teleinformatics 79,* Boutmy and Danthine (eds), North Holland Publishing Company, 1979.

Lanning, L. J. and Leonard, M. S. "File Allocation in a Distributed Computer Communication Network," *IEEE Transactions on Computing,* Volume C-32, Number 3, 1983, pp. 232-244.

Mahmoud, S. and Riordan, J. S. "Optimal Allocation of Resources in Distributed Information Networks," *ACM Transactions on Database Systems,* Volume 1, Number 1, 1976, pp. 66-78.

Morgan, H. L. and Levin, K. D. "Optimal Program and Data Locations in Computer Networks," *Communications of the ACM,* Volume 20, Number 5, 1977, pp. 315-321.

Pirkul, H. "An Integer Programming Model For the Allocation of Databases in a Distributed Computer System," *European Journal of Operational Research,* forthcoming (1986).

Ramamoorthy, C. V. and Wah, B. W. "The Isomorphism of Simple File Allocation," *IEEE Transactions on Computing,* Volume C-32, 1983, pp. 221-231.

TYMNET, *TYMNET The Intellegent Network People,* Tymnet Inc., Great Lakes District, Pittsburgh, Pensylvania, (1980).

Urano, U., Ono, K. and Inoue, S. "Optimal Design of Distributed Networks," *Second International Conference on Computer Communications,* 1974, pp. 413-420.

Whitney, V. K. M. "A Study of Optimal File Site Assignment and Communication Network Configuration in Remote Access Computer Message Processing and Communications Systems," Ph.D. Dissertation, University of Michigan, 1970.