ANALYSIS OF CONCURRENCY IN DATA BASE SYSTEMS

by

FERNANDO ISAAC SILVA LOPEZ

B. S., Escuela Superior de Ingenieria Mecanica y Electrica
Instituto Politecnico Nacional (Mexico), 1974

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

Approved by:

Major Professor

INDEX

INDEX (continued)

I. Introduction

Considerable skill is required to use current computer systems today in order to handle and process high volumes of information at maximum speed and minimum cost. It has been demonstrated (8) that the systems architecture and the data-base characteristics highly determine those two factors. Since data-bases and system architectures are very important elements in our highly computerized society, a methodology is necessary to evaluate their characteristics in order to determine the most attractive architecture in terms of economics and performance. This methodology may be developed in several ways. For example, if for someone it is very important to obtain fast response time to queries, we must consider the type of DBMS to be utilized as well as the physical characteristics of the data-base. In addition, the achievement of concurrency through a network of distributed processors should be considered. On the other hand, if for someone it is more important to save money rather than to obtain a fast response time then s/he would probably be little interested in the use of concurrency but would pay much attention the data-base characteristics since they greatly determine the cost. In order to provide a solution the user must balance his requirements between response time and cost.

In this study we have tried to provide a broader set of data-bases than those used by Slonim (8) in order to have a wider set of results which can serve as basis for future work in this area.

Our goals in this study were to determine:

1) How changes in the size of the data-base affect the response time and cost? and,

2) How changes in the record length affect the response time and cost?

The response time and cost were calculated using two forms of processing, centralized and concurrent.

II. Background

This study is a continuation of work performed independently by Cardenas (1) and Slonim (8). Cardenas (1) analyzes the performance of centralized DBMS architecture. In his research Cardenas analyzes a model of an inverted file organization to obtain estimates of average retrieval time and total storage performance. The models that he describes take into account storage structure and search of the indexes, statistics of contents of the data-base, logical complexity of the queries and macroscopic device parameters. Cardenas quantifies the influence of these factors upon the performance of data base systems.

The research carried out by Slonim (8) is related to the use of the GDIMA (Generalized Distributed Information Management Architecture) architecture. Slonim's model makes use of all the parameters used by Cardenas plus new parameters which make possible the use of concurrent processing. Both Cardenas and Slonim worked with the average access time for retrieval using four kinds of query complexities. Slonim also considered the cost incurred in the use of distributed processing against centralized processing.

III. The Models

The models used in this study are those models described by Cardenas (1) and Slonim (8). The equations for concurrent processing developed by Slonim have been modified slightly in order to provide a closer correspondence to equations of Cardenas. In this section we are going to describe the set of equations developed by Cardenas for centralized processing and the set of modified equations used for concurrent processing. Finally the set of equations used for cost will be described.

Appendix A presents several tables describing all the parameters used in the different sets of equations described in this section.

The structure of the logical I/O of an information system is determined by the type and amount of logical storage required. These factors can not be fixed, but can be estimated according to a series of equations developed by Cardenas (1). His equations are limited to inverted files. Slonim (8) chose to compare sequential with concurrent processing in terms of their performance on inverted files. The reliability of Cardenas' equations has been confirmed by others researchers in the field (5).

3.1 Storage Parameters (Estimated)

(a) Fixed key-value length (KLFIX): specifies an average fixed length for key-values.

$$KLFIX = \left\lceil \frac{KLAVE}{WORD} \right\rceil_{+} \qquad (3.1)$$

where WORD is the word-length for a particular computer, and KLAVE is the average key-value length over the inverted keys used to query the data-base. The "plus" indicates that the result of the division is rounded to the next higher integer. The "minus" is going to indicate that the result is truncated to the next lower integer.

(b) Blocking Factor (Key-value Index Block).

$$BFI = \left[\frac{BLOCKW}{KLFIX + 2}\right]_{-} \qquad (3.2)$$

where BLOCKW is the Block-length in words.

(c) Reserve Spaces (Key-value Index Block). 10 percent reserve-spaces for key-value index-block

$$RSI = \left[\frac{BFI}{10}\right]_{+} \qquad (3.3)$$

the choice of 10 percent is arbitrary but convenient. An environment in which the volumes of insertions is high may demand more than 10 percent.

(d) Storage Requirement (Key-value Index Value). The number of key-value blocks required to store the index-block is indicated by Eq. 3.4:

$$SRI = \left[\frac{\sum_{I=1}^{NKEY} NVAL(I)}{BFI - RSI}\right]_{+} \qquad (3.4)$$

where NVAL(I) is the number of different key-values associated with each key-name (I), and NKEY is the number of key-names in which inversion is carried out (i.e., the number of inverted key-names).

(e) Blocking Factor (Accession Block).

$$BFA = BLOCKW \qquad (3.5)$$

The BFA is the blocking-factor for accession pointers.

(f) Reserve Spaces (Accession Block).

$$RSA = \left[\frac{BFA}{10} \right]_{+} \tag{3.6}$$

(10 percent reserve-spaces for accession-pointers).

(g) Storage Requirement (Accession Block). The number of accession pointer-blocks required is given by Eq. 3.7:

$$SRA = \left[\frac{NKEY * NREC}{BFA - RSA} \right]_{+} \tag{3.7}$$

where NREC is the total number of records in the data-base.

(h) Blocking Factor (Data Block).

$$BFD = \left[\frac{BLOCKC}{RLAVE} \right]_{-} \tag{3.8}$$

where BLOCKC is the block-length in bytes and RLAVE is the average key-value length for the number of access-key names (i.e., NKEY).

(i) Reserve Spaces (Data Block).

$$RSD = \left[\frac{BFD}{10} \right]_{+} \tag{3.9}$$

(10 percent reserve-spaces for data-records).

(j) Storage Requirement (Data Blocks). The average space required for the data-records is expressed by Eq. 3.10:

$$SRD = \left[\frac{NREC}{BFD - RSD} \right]_{+} \tag{3.10}$$

(k) Total Storage Requirement. The total average storage requirement for the inverted organization is expressed by Eq. 3.11:

$$TOTSR = 1 + SRI + SRA + SRD \tag{3.11}$$

3.2 User Requirements

There are four complexities of query. Each of these requires a separate algorithm for the calculation of average response-time. These four algorithms are dependent upon the storage parameter equations given above, the I/O parameters, storage devices, and the following three equations.

The first establishes the average list-length for a query.

$$LSTAVE = \left[\frac{NKEY * NREC}{\sum_{I=1}^{NKEY} NVAL(I)} \right] \tag{3.12}$$

where NKEY is the number of access-key names inverted and NREC is the total number of records. NVAL(I) is the number of unique key-values for the (I)th key.

The second equation establishes the number of data-blocks accessed by a single query.

$$X_D = SRD*(1-(1-(1/SRD))**K) \tag{3.13}$$

This expression gives the average number of blocks X_D which contain the K records to be retrieved, where SRD is the storage requirement for data-blocks, and K characterizes the query-complexity.

The third equation establishes the average number of accession-blocks required to hold the average length-list (i.e., LSTAVE).

$$X_A = \left[\frac{LSTAVE}{BFA} \right]_+ \tag{3.14}$$

In order to compute average access time, estimates of three basic units of time are needed:

1. T_T, the average time to access a block or page (e.g., a track).

2. T_I, the average time to intersect two blocks of accesion numbers or pointers to records.

3. T_C, the average time to compare an access key or key-value.

These estimates depend of course on specific device environments. As indicated in table A.II the values used here are: $T_T = 100$ msec, $T_C = 1.5$ msec, and $T_I = 50$ msec. These values can be adjusted to describe the physical characteristics of any secondary storage devices.

The search strategy, and consequently, the average access time are a function of the complexity of the retrieval requirements. The parameters ACI, ICR and RCQ are supplied by the user or, preferably, should be obtained from measurements of the incoming queries.

3.3 Algorithms for Centralized Processing. (1)

Complexity 1: the query demands that the atomic condition A will have the form

$$\text{Item-name} \left\{ \begin{matrix} < \\ = \\ \neq \\ > \end{matrix} \right\} \text{Value}$$

where Item-name is the key-name or attibute-name in the COBOL-record sense, or the domain in the tubular or relational sense

Algorithm 1:

Step 1. Read track-index: T_T

Step 2. Search track-index: T_C*Log_2SRI

Step 3. Read key-value block: T_T

Step 4. Search key-value block: T_C*Log_2BFI

Step 5. Read accession pointer list:

$$T_{T}* \left[\frac{LSTAVE}{BFA} \right]_{+}$$

Step 6. Read data blocks: $T_T * X_D$

Equation 3.15 represents the average access-time for a query of Complexity 1.

$$ACCTM = T_C \, (Log_2 \, SRI + Log_2 \, BFI) + T_T \left(2 + \left[\frac{LSTAVE}{BFA} \right]_{+} + X_D \right) \quad msec \tag{3.15}$$

where X_D = SRD*(1-(1-(1/SRD))**K)

K = LSTAVE

Complexity 2: An item-condition A is a disjunction of atomic conditions A_1 or A_2 or .... A_i .... A_j, such that each A_i reflects the same item name (key-name or domain). ACI is defined as the number of atomic conditions per item condition I.

Example: Age 20 or Age 21, where ACI 2

Algorithm 2:

Step 1. Read track index: T_T

Step 2. Search track index block: T_C * Log_2 SRI

Step 3. Read key-value block: T_T

Step 4. Search key-value block: T_C * Log_2 BFI

Step 5. Repeat step 4 for each of the ACI atomic conditions (key-names)

Step 6. Read and merge (OR) accession lists:

$(T_I + T_T)$ * ACI * X_A

Step 7. Read data blocks: T_T * X_D

Equation 3.16 represents the average access-time for a query of Complexity 2

$$ACCTM = T_C\ (Log_2\ SRI + (ACI * Log_2 BFI)) +$$

$$T_T\ (2 + ACI * X_A + X_D) + T_I * (ACI * X_A)\ \text{msecs} \qquad (3.16)$$

where K = ACI * LSTAVE

Complexity 3: A record-condition R is a conjunction of item conditions I_1 and I_2 and ... I_m ... and I_n such that each I reflects a distinct item-name. ICR is defined as the number of item conditions per record condition R.

Example: (Age > 20) and (Sex = Female)

where ICR = 2, ($ACI_1 = ACI_2 = 1$).

Algorithm 3:

Step 1. Read track index: T_T

Step 2. Search track index block: $T_C * Log_2$ BFI

Step 3. Read key-value block: T_T

Step 4. Search key-value block: $T_C * Log_2$ BFI

Step 5. Repeat step 4 for each of the ACI atomic conditions.

Step 6. Repeat step 2 through 5 for each of the ICR item conditions.

Step 7. Read, merge (OR) and intersect (AND) accession lists:

$$\left(\sum_{i=1}^{ICR} ACI_i \right) * X_A\ (T_T + T_I)$$

Step 8. Read data blocks: $T_T * X_D$

Equation 3.17 represents the average access-time for a query of Complexity 3.

$$ACCTM = T_C * ICR (Log_2 SRI + (ACI * Log_2 BFI)) +$$

$$T_T (1 + ICR + X_D) + \left(\sum_{i=1}^{ICR} ACIi \right) * X_A (T_T + T_I) \quad (3.17)$$

where K LSTAVE * ACI

Complexity 4: A query-condition Q, is a disjunction of record conditions, R_1 or R_2 or .... R_i ... or R_j. RCQ is defined as the number of record conditions per query condition Q.

Example: [(Age > 18) and (Sex = Female)] or [(Age > 20) and (Sex = Male) and (Class = Senior or Class = Graduate)]

Algorithm 4:

Step 1. Read track index: T_T

Step 2. Search track index block: $T_C * Log_2 SRI$

Step 3. Read key-value block: T_T

Step 4. Search key-value block: $T_C * Log_2 BFI$

Step 5. Repeat step 4 for all atomic conditions reflecting the same key-name.

Step 6. Repeat steps 2-5 for each one of the different key-names appearing in the query.

Step 7. Read, merge (OR) and intersect accession lists:

$$\left(\sum_{i=1}^{ICR} ACIi \right) * RCQ * X_A (T_T + T_I)$$

Step 8. Read data blocks: $T_T * X_D$

Equation 3.18 represents the average access-time for a query of Complexity 4.

$$ACCTM = T_C * ICR\ (Log_2\ SRI \quad (ACI * RCQ * Log_2\ BFI))$$

$$+\ T_T\ (1 + ICR + X_D) + \left(\sum_{i=1}^{ICR} ACI \right) * RCQ * X_A\ (T_T + T_I) \quad \text{msec} \qquad (3.18)$$

where $K = \left(\sum_{j=1}^{RCQ} (ACI)_j \right) * LSTAVE$

3.4 Algorithms for concurrent processing.

These algorithms are based on the ones developed by Slonim (8) for the use of the GDIMA architecture. In these algorithms various parts of the data base processing are distributed to multiple microprocessors.

Complexity 1:

Step 1. Store track-index, key-value block and accession-pointer list in a microprocessor: T_T

Step 2. Search track-index: $T_C * Log_2\ SRI$

Step 3. Search key-value block: $T_C * Log_2\ BFI$

Step 4. Read data-blocks and use as many microprocessors as needed from processor-pool: $T_T * X_D/P$

Equation 3.19 represents the average access-time for a query of complexity 1.

$$ACCTM = T_C * (Log_2\ SRI + Log_2\ BFI) + (T_T * (3 + X_D/P)) \qquad (3.19)$$

where P is the number of processors required for input/output. K and X_D are estimated in the same way that were estimated for centralized processing. In a query of this complexity, the concurrent processors work only on the I/O.

Complexity 2:

Step 1. Store track-index and key-value in microprocessor: T_T

Step 2. Search track-index block: T_C * Log_2 SRI

Step 3. Search key-value block: T_C * Log_2 BFI

Step 4. Assign to each ACI a microprocessor and repeat step 3 for each of the ACI: $T_C * \left(\frac{ACI}{CACI} * Log_2\ BFI \right)$

Step 5. Merge accession-lists: $\frac{ACI}{CACI} * (T_T + T_I)$

Step 6. Read data-block and use as many processors as needed from processor pool: $T_T * \frac{X_D}{P}$

Equation 3.20 represents the average access-time for a query of complexity 2.

$$ACCTM = T_C \left(Log_2\ SRI + \left(\left(\frac{ACI}{CACI} \right)_+ * Log_2\ BFI \right) \right) +$$

$$T_T \left(2 + \frac{ACI}{CACI} * X_A + \frac{X_D}{P} \right) + T_I \left(\frac{ACI}{CACI} \right)_+ * X_A \qquad (3.20)$$

where CACI is the number of concurrent processors assigned to an item-condition.

$$X_D = SRD\ (1-(1-(\frac{1}{SRD}))^K)$$

where K = ACI * LSTAVE

In this complexity of query the concurrent processors work on an item-condition and on the I/C.

Complexity 3:

Step 1. Assign to each ACI a microprocessor: CACI

Step 2. Assign to each ICR a microprocessor: CICR

Step 3. Store track-index and key-value in a microprocessor: T_T

Step 4. Search track-index block for each CICR:

CICR (T_C * Log_2 BFI)

Step 5. Search key-value block for each CACI for each of the CICR: CICR (CACI * T_C * Log_2 BFI)

Step 6. Read, merge (OR) and intersect (AND) accession list:

$$T_T \left(\sum_{i=1}^{CICR} CACIi \right) * T_I * X_A$$

Step 7. Read data-block and use as many microprocessors as needed from processor pool: $T_T * X_D/P$

Equation 3.21 represents the average access-time for a query of complexity 3.

$$ACCTM = T_C \left(\frac{ICR}{CICR} \right)_+ * \left(Log_2 \; SRI + \left(\frac{ACI}{CACI} \right)_+ * Log_2 \; BFI \right) +$$

$$T_T \left(1 * \left(\frac{ICR}{CICR} \right)_+ + \left(\frac{X_D}{P} \right) \right) + \left(\frac{ACI}{CACI} \right)_+ * \; ICR * \left(X_A \; * \left(T_T + T_I \right) \right) \tag{3.21}$$

where CICR is the number of concurrent processors assigned to an ICR. X_D and K are calculated in the same way that was done for complexity 2.

Complexity 4:

Step 1. Assign to each ICR a microprocessor: CICR

Step 2. Assign to each ACI a microprocessor: CACI

Step 3. Assign to each RCQ a microprocessor: CRCQ

Step 4. Store track-index and key-values in microprocessor: T_T

Step 5. Search track-index block: T_C * (Log_2 SRI) * CICR

Step 6. Search key-value block:

T_C * (CACI(CRCQ(Log_2 BFI))) * CICR

Step 7. Read, merge (OR) intersect accession list:

$$T_I * \left(\sum_{i=1}^{CICR} CACIi * CRCQ * X_A \right)$$

Step 8. Read data-block and use as many processors as required from processor pool: $T_T * \frac{X_D}{P}$

Equation 3.22 represents the average access-time for a query of complexity 4.

$$ACCTM = T_C \left(\frac{ICR}{CICR} \right)_+ * \left(Log_2\ SRI \left(\frac{ACI}{CACI} \right)_+ * \left(\frac{RCQ}{CRCQ} \right)_+ * Log_2\ BFI \right) +$$

$$T_T \left(1 + \left(\frac{ICR}{CICR} \right)_+ + \left(\frac{X_D}{P} \right) \right) +$$

$$\left(\left(\frac{ACI}{CACI} \right)_+ * CICR \right) * \left(\frac{RCQ}{CRCQ} \right)_+ * X_A * \left(T_T + T_I \right) \qquad (3.22)$$

where K = ACI * RCQ * LSTAVE

X_D = SRD * (1-(1-(1/SRD))**K)

and CRCQ = number of concurrent processors assigned to RCQ.

In this complexity of query the concurrent processors work on record-condition query and item-condition record and a condition item and on the I/O.

3.5 Algorithms for Cost

In a normal working environment, cost is as important as access time. Slonim (8) gives great importance to this point and he describes the equations that must be used to calculate the cost for centralized and concurrent processing. For CPU-cost in the centralized system Slonim used prices charged by the Kansas State University computer center (6). For the microprocessors he imposed a single price for all varieties of processor ($5,000 each) such price includes development-costs and the manufacture's profit.

Equations 3.23 through 3.27 (8) provide a means of calculating such costs for both sequential and concurrent processing.

Equation 3.23 gives the CPU-cost for sequential processing on a large centralized machine (ITEL AS5 at Kansas State University):

$$\text{CPU SEQ. COST} = \left[(9¢ \text{ PER SECOND}) * (\text{ACCESS TIME/SEC}) \right] + \left[(0.037¢ * \text{CORE SIZE/SEC}) * (\text{ACCESS TIME/SEC}) \right] \quad (3.23)$$

Equation 3.24 is a general equation for calculating cost-per-second on a microcomputer:

$$\text{COST PER SEC.} = \left[(\text{NO. OF YEARS}) * (\text{NO. OF MONTHS}) * (\text{NO. OF DAYS}) * (\text{NO. OF SEC}) \right] \Big/ \left[(\text{NUMBER OF PROCESSORS}) * (\text{COST PER PROCESSOR}) \right] \quad (3.24)$$

Equation 3.25 provides the cost of CPU in concurrent processing:

$$\text{COST OF CONC. CPU/SEC} = (\text{COST PER SEC}) * (\text{CONC. ACCESS TIME}) \quad (3.25)$$

Equation 3.26 calculates storage-costs. It is assumed that both the sequential and concurrent processors use the same storage-devices:

$$\text{STORAGE COST/SEC} = (\text{COST PER TRACK}) * (\text{NUMBER OF TRACKS}) * (\text{ACCESS TIME}) \quad (3.26)$$

Equation 3.27 provides the total cost of processing:

$$\text{TOTAL COST/QUERY} = \text{CPU COST} + \text{STORAGE COST} \quad (3.27)$$

The values used in this report for the parameters of these equations are shown in Appendix A.

In this chapter we have described equations for storage-requirements, access-times and costs for centralized and concurrent processing. Equations 3.1 through 3.11 described the storage-requirements needed in the processing of data-bases working under centralized or concurrent architectures. These equations are exercised in next chapter to calculate the storage requirements for the five groups of data-bases under study. The equations 3.15 through 3.18 described the four query complexities for centralized processing. Equations 3.19 through 3.22 described the equations developed in this study for concurrent processing. The reason of the development of this new set of equations was to have a group of equations which could be easily related to the equations given by Cardenas (1) for centralized processing. Finally equations 3.23 through 3.27 described the equations used for calculating both centralized and concurrent costs. In Chapter IV all the equations described here are exercised and the results and analysis of the experiments are given in Chapter V.

IV. The Experiment

In this section we describe the data-bases used in this study and the cost and performance experiments performed on the data-bases.

4.1 Parameterization

The first step carried out in our experiments was to determine the values for storage requirements using the equations 3.1 through 3.11. After that we proceed to compute the access-time for centralized processing for the different levels of complexity using the characteristics for each data-base utilized by Slonim (8) by means of equations 3.15 through 3.18 then the concurrent processing performance equations (Eq. 3.19-3.22) were exercised on the various data-bases. The next step was to use the cost equations (Eq. 3.23-3.27) to compare the costs incurred in centralized and concurrent processing. To finish with our experiments we calculated the improvement of response-time centralized against concurrent processing. Likewise the improvement in cost is calculated and the results are given in percentage.

4.2 Description of the Data-bases

Cardenas (1) in his research worked with 6 data bases, each with different characteristics, the experiments were carried out on inverted files. Such files are of the structured class, and the validity of the experiments is therefore somewhat limited. On the other hand, inverted files are common to DBMS and more generalized information systems. Moreover, Cardenas selected inverted files because he found that performance on these was indicative of performance on other type of files. The experiments carried out in this report have been restricted to inverted files only because of a lack of comparative statistics for other types of files. In this study we use 8 data-bases as did Slonim (8). All of these data-bases have a single structure (i.e., inverted file), but the contents differ in each case. The results of the experiments are analyzed according to four factors:

i) Storage Structure: the storage requirements for the value-index block, the access block, the data-block, and the sum of these requirements.

ii) Access-time and cost for the four query complexities.

iii) Access-time for the fourth query-complexity at various levels of concurrency.

iv) Costs for the fourth query-complexity at various levels of concurrency.

4.2.1 System-Statistics and Parameters used

Throughput is determined in large part by design constraints designed by the user. Slonim (8) defines five standard categories of response time: "excellent, or almost instantaneous; very good, less than two seconds; good, two to four seconds; average, greater than four seconds but less than fifteen; and poor, greater than fifteen seconds".

Another decisive factor which determines throughput is the size of the data-base. In our experiments we use several groups of data: the first group which we call "Normal" has the same characteristics used by Cardenas and Slonim; the second group which we call "Small" has a change in the number of records (NREC) of each data-base, we used 800 records in each data-base remaining without change the other parameters; the third group which we call "Large" uses 51000 records in each data-base remaining without change the other parameters; the fourth group which we call "Short" has a record length of 20 characters in each data-base remaining its number of records and other characteristics equal to the Normal group; the last group is called "Long" and it has a record length of 300 characters in each data-base again remaining the number of records and other characteristics equal to the Normal group. The value of the parameters choosen for the last four groups were determined upon previous experience. Although is well known that a "Small" data-base can be smaller than 800 records or a "Large" data-base can be composed of million of records, however, we decide to work with the mentioned values, since they are well within the range of existing data bases.

A final factor influencing throughput is the frequency of queries of each level of complexity.

Tables A.II and B.I show the characteristics of the data-bases used by Cardenas. Those same characteristics and values were used in this study in order to have a starting point to work with. The values and characteristics used for data-bases 7 and 8 were obtained from the set of data bases described by Slonim (8).

Summary of the characteristics and results of the five groups ("Normal", "Small", "Large", "Short", and "Long") are given in Appendix B.

In this Chapter we described how the experiment was carried out. First of all we determined the storage-requirements for the five groups of data-bases. Then we proceed to compute the access-time for the four query complexities for centralized processing using the five groups of data-bases described. After that was computed the access-time for the four query complexities for concurrent processing at four levels of concurrency. At the first level 8 microprocessors were added to deal with I/O functions. At the second level 4 microprocessors more were added to deal with atomic-conditions. At the third level 20 microprocessors were added to deal with item-conditions and at the fourth level of concurrency 40 microprocessors more were added to deal with record-conditions. Access-times at the four levels of concurrency were calculated for the five groups of databases. Finally the cost incurred in centralized and concurrent processing was computed.

V. Results and Analysis

In this section an analysis of the results obtained with the experiments carried out in section IV is presented.

5.1 Analysis of Storage-Structures

Tables 5.1 through 5.5 list storage requirements for each group of data-bases, calculated according to equations 3.1 through 3.11. Figures 5.1 through 5.5 present the totals in graphic form. The meaning for all the parameters and symbols utilized in this chapter are given in Appendix A.

A) "Normal" Group: (see tables 5.1 and B.1 and figure 5.1) Data-base 1 requires more tracks for the index-block than the others data-bases, the reason is that it has longer key-values length. Data-base 2 requires less total storage requirements than data-base 1, although they have the same number of records data-base 2 uses much less tracks for the index-block. Data-base 3 utilizes the same storage for index-block and accession-block that data-base 2 and the storage for data-blocks is bigger in data-base 3 than data-base 2 since data-base 3 has more records giving as result more total storage required. Data-base 4 is the largest data-base in this group since it has much more number of records than the other data-bases, however, its storage requirements are not the larger, this is due to the fact that its record-length is the shorter of the group. Data-base 5 has an important increment of storage requirements with respect to data-bases 1, 2, 3 and 4 this due to the fact that this data-base is the second largest data-base utilizing a record length

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

DATA BASE	KLFIX	BFI	RSI	SRI	BFA	RSA	SRA	BFD	RSD	SRD	TOTSR
1	2	447	45	8	1790	179	14	123	13	34	57
2	1	596	60	1	1790	179	10	123	13	34	46
3	1	596	60	1	1790	179	14	115	12	51	67
4	1	596	60	3	1790	179	47	127	13	163	214
5	1	596	60	1	1790	179	60	45	5	398	460
6	1	596	60	1	1790	179	4	26	3	57	63
7	1	596	60	1	1790	179	4	45	5	33	39
8	1	596	60	1	1790	179	60	26	3	691	753

TABLE 5.1

Total Storage Requirements for DBMS 1 through 8

("Normal" Group)

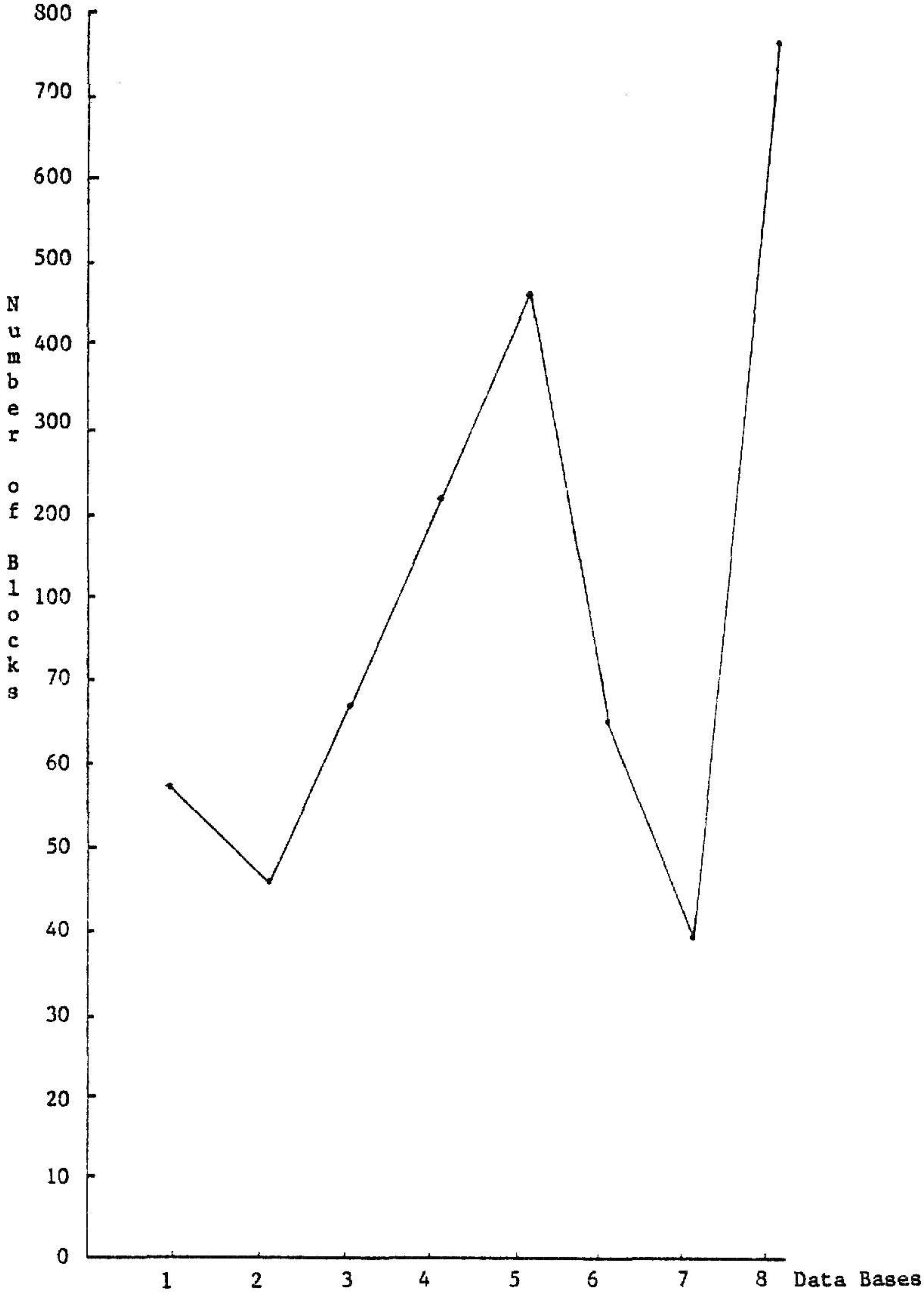

Figure 5.1

Graphic Representation of the Total Storage Requirements for "Normal" Group much bigger than the others data-bases. Data-base 6 utilizes the smallest number of records in the group but the bigger record length those factors give as result a relative small storage requirement. Data-base 7 has the same number of records that data-base 6 but it uses a shorter record length giving as result the smallest storage requirements of the group. Data-base 8 utilizes the largest storage requirements among the set of data bases this is due to the fact that it is a large data base combined with a long record length.

B) "Small" Group: (see tables 5.2 and B.2 and figure 5.2) Data-base 1 utilizes the bigger storage for index-block this is due to the fact that its number of distinct key-values and average key-value length are the greatest in the set, however, its storage required for data-block ranges in the lower levels since its record length is one of the lower lengths. Data-base 2 utilizes the minimum total storage requirements the reasons are that the number of distinct key-values, the average key-value length and record-length are small compared with data-base 1. Data-base 3 and 4 utilizes almost the same storage requirements that data-base 1 does, the analysis can be the same one done for data-base 1. Data-base 5 and 7 utilize almost same storage requirements since both are similar data-bases their total storage required are bigger than data-bases 2, 3 and 4 because the record length utilized by data-bases 5 and 7 is greater than the others. Data-bases 6 and 8 utilize the maximum total storage required for this group the reason is that they have the longer record length among the data-bases.

C) "Large" Group: (see tables 5.3 and B.3 and figure 5.3) This group in general presents the largest storage requirements

DATA BASE	KLFIX	BFI	RSI	SRI	BFA	RSA	SRA	BFD	RSD	SRD	TOTSR
1	2	447	45	8	1790	179	3	123	13	8	20
2	1	596	60	1	1790	179	2	123	13	8	12
3	1	596	60	1	1790	179	2	115	12	8	12
4	1	596	60	3	1790	179	2	127	13	8	14
5	1	596	60	1	1790	179	3	45	5	20	25
6	1	596	60	1	1790	179	2	26	3	35	39
7	1	596	60	1	1790	179	2	45	5	20	24
8	1	596	60	1	1790	179	3	26	3	35	40

TABLE 5.2

Total Storage Requirements for DBMS 1 through 8

("Small" Group or NREC = 800)

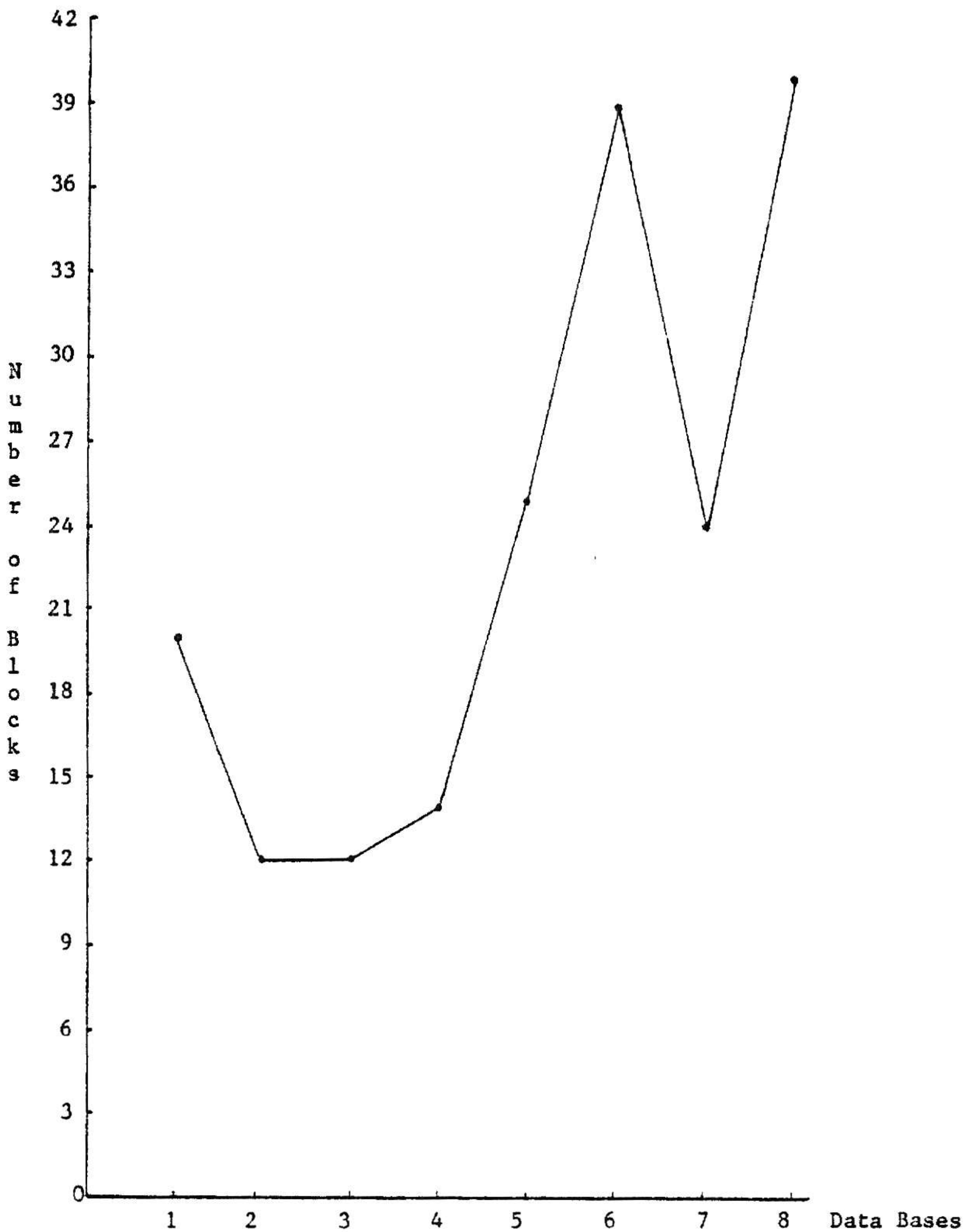

Figure 5.2

Graphic Representation of the Total Storage Requirements for "Small" Group among the five groups the reason is that this group utilizes the greatest number of records. Data-base 1 as in the last two groups continues with the greatest storage requirements for index-blocks since it contains the greatest number of distinct key-values and key-value length, the storage for accession blocks ranges in the higher levels this is due to the fact that the number of inverted keys for this data-base is greater than for the other data-bases. Data-base 2 presents a decrease in the total storage this is due to the number of inverted-keys which are smaller in this data-base than data-base 1. Data-base 3 has an increment in the storage for data-blocks since the record length for this data base is sligthly longer than data-base 1 and 2 which gives a larger total storage. Data-base 4 presents an increment of storage for index-blocks, the storage for accession blocks remains as in data-base 2 and 3 but it has a decrement in storage for data-blocks giving as result the minimum total storage requirements for this group. The reason of the decrement in storage for accession blocks is that the length of the record is the smaller in the group. Data-base 5 has increments of accession-block storage and data-blocks storage, this is because there exist increments in the number of inverted keys and in the record length of this data-base. Data-base 6 has a big increment of total storage requirements due to the fact that the record length is the longer in the group. Data-base 7 has a decrement in the total storage since the record length has also a decrement. Data-base 8 utilizes more storage than the other data bases. The reason is that it has increments in the storage for accession-blocks and the storage for data-block remains at the same level that data-base 7.

DATA BASE	KLFIX	BFI	RSI	SRI	BFA	RSA	SRA	BFD	RSD	SRD	TOTSR
1	2	447	45	8	1790	179	190	123	13	464	663
2	1	596	60	1	1790	179	127	123	13	464	593
3	1	596	60	1	1790	179	127	115	12	496	625
4	1	596	60	3	1790	179	127	127	13	448	579
5	1	596	60	1	1790	179	190	45	5	1275	1467
6	1	596	60	1	1790	179	127	26	3	2218	2347
7	1	596	60	1	1790	179	127	45	5	1275	1404
8	1	596	60	1	1790	179	190	26	3	2218	2410

TABLE 5.3

Total Storage Requirements for DBMS 1 through 8

("Large" Group or NREC = 51000)

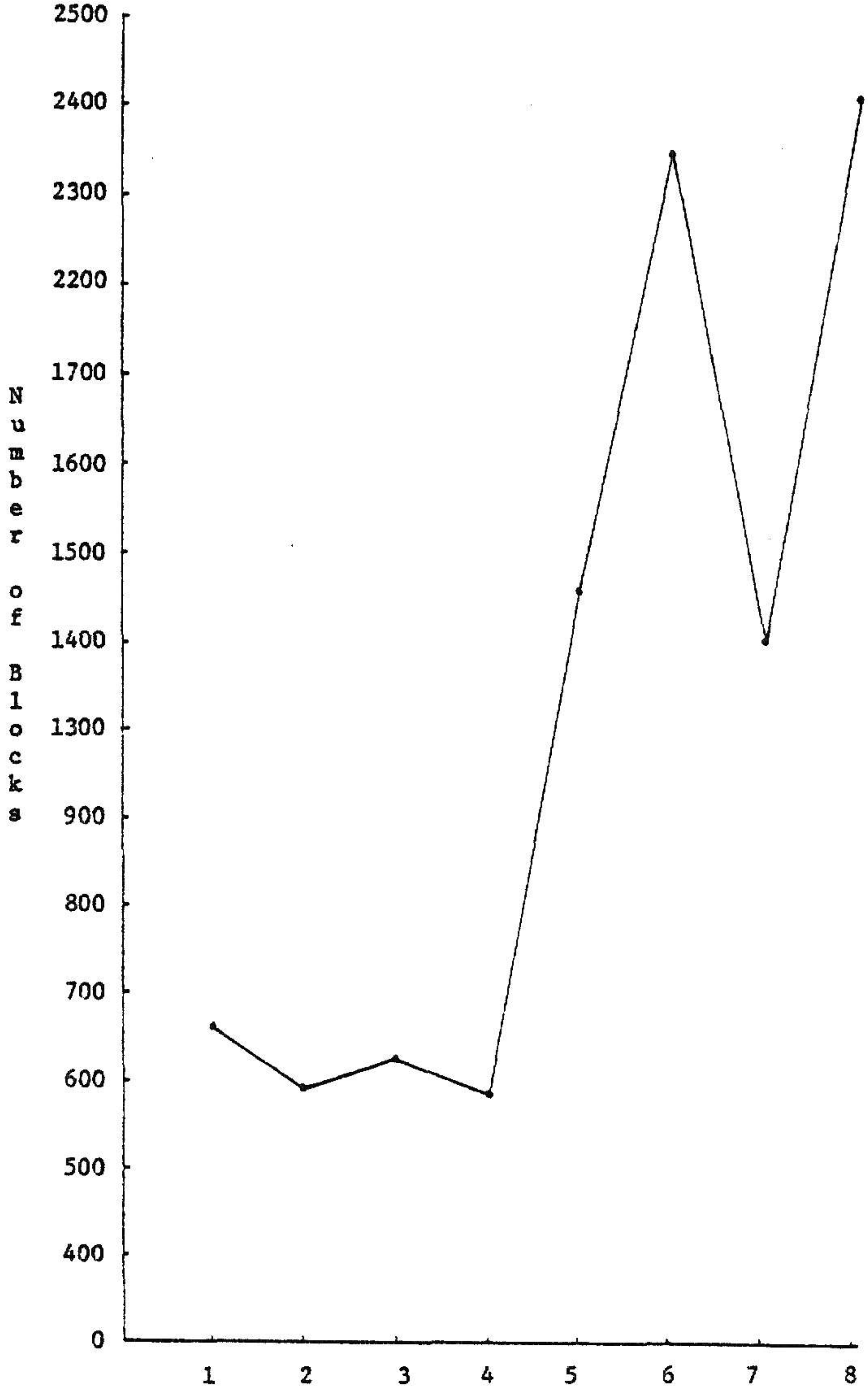

Figure 5.3

Graphic Representation of the Total Storage Requirements for "Large" Group

D) "Short" Group: (see tables 5.4 and B.4 and figure 5.4) Data-base 1 utilizes a high number of tracks for index-block due to its high value of key-value length and high number of distinct key-values, storage for accession blocks and data-blocks is not high since it is a medium data-base in this group. Data-base 2 utilizes less storage for accession-block than data-base 1. The reason for this is that the number of inverted keys is less in this data-base; therefore, the total requirements for storage are less. Data-base 3 has an increment in the total requirements. Data-base 4 has increments of storage in index-block, accession-block and data-block due to the fact that this data-base utilizes a greater number of distinct key-values than do data-base 2 and 3. In addition, the number of records is the higher of the group all this gives as result an increment in the total storage requirements. Data-base 5 and 8 are similar in characteristics therefore their storage requirements are the same, both of them utilize the maximum amount of total requirements in this group, this is because the number of inverted keys combined with the high number of records they have given as result a large total storage requirements. Data-base 6 and 7 utilize the minimum total storage requirements. This is due to the fact that both of them are small data-bases with small number of inverted keys.

E) "Long" Group: (see tables 5.5 and B.5 and figure 5.5) Data-base 1 utilizes more number of tracks for index-block than other data-bases due to its high value of key-value length and high number of distinct key-values, the average list-length is the minimum since its number of distinct key-values is the higher in the group. Since data-base 2 has a reduction in its number of inverted keys this

DATA BASE	KLFIX	BFI	RSI	SRI	BFA	RSA	SRA	BFD	RSD	SRD	TOTSR
1	2	447	45	8	1790	179	14	537	54	8	31
2	1	596	60	1	1790	179	10	537	54	8	20
3	1	596	60	1	1790	179	14	537	54	11	27
4	1	596	60	3	1790	179	47	537	54	39	90
5	1	596	60	1	1790	179	60	537	54	33	95
6	1	596	60	1	1790	179	4	537	54	3	9
7	1	596	60	1	1790	179	4	537	54	3	9
8	1	596	60	1	1790	179	60	537	54	33	95

TABLE 5.4

Total Storage Requirements for DBMS 1 through 8

("Short" Group or RLAVE = 20)

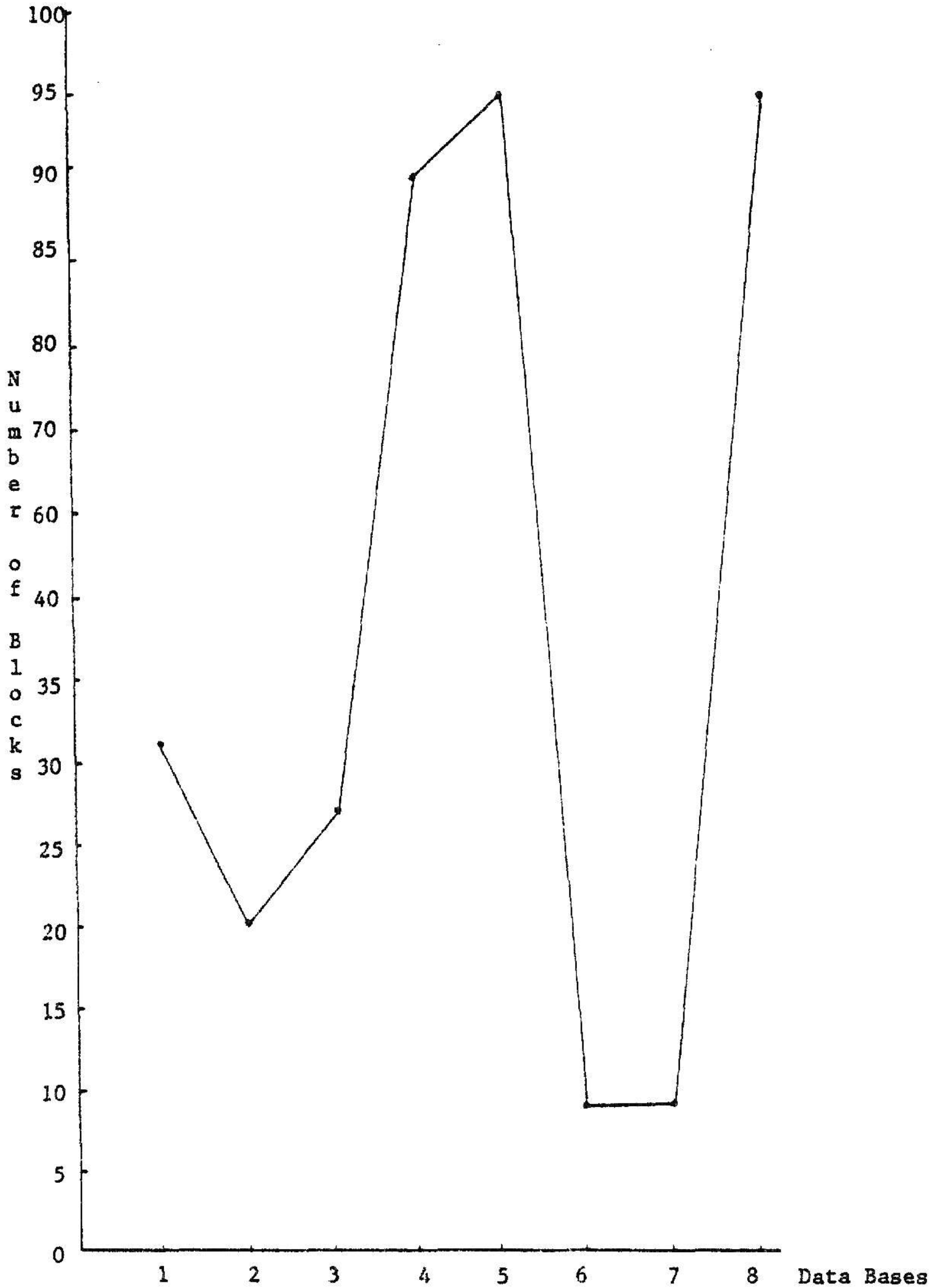

Figure 5.4

Graphic Representation of the Total Storage Requirements for "Short" Group

DATA BASE	KLFIX	BFI	RSI	SRI	BFA	RSA	SRA	BFD	RSD	SRD	TOTSR
1	2	447	45	8	1790	179	14	13	2	335	358
2	1	596	60	1	1790	179	10	13	2	335	347
3	1	596	60	1	1790	179	14	13	2	447	493
4	1	596	60	3	1790	179	47	13	2	1689	1740
5	1	596	60	1	1790	179	60	13	2	1445	1507
6	1	596	60	1	1790	179	4	13	2	118	124
7	1	596	60	1	1790	179	4	13	2	118	124
8	1	596	60	1	1790	179	60	13	2	1445	1507

TABLE 5.5

Total Storage Requirements for DBMS 1 through 8

("Long" Group or RLAVE =800)

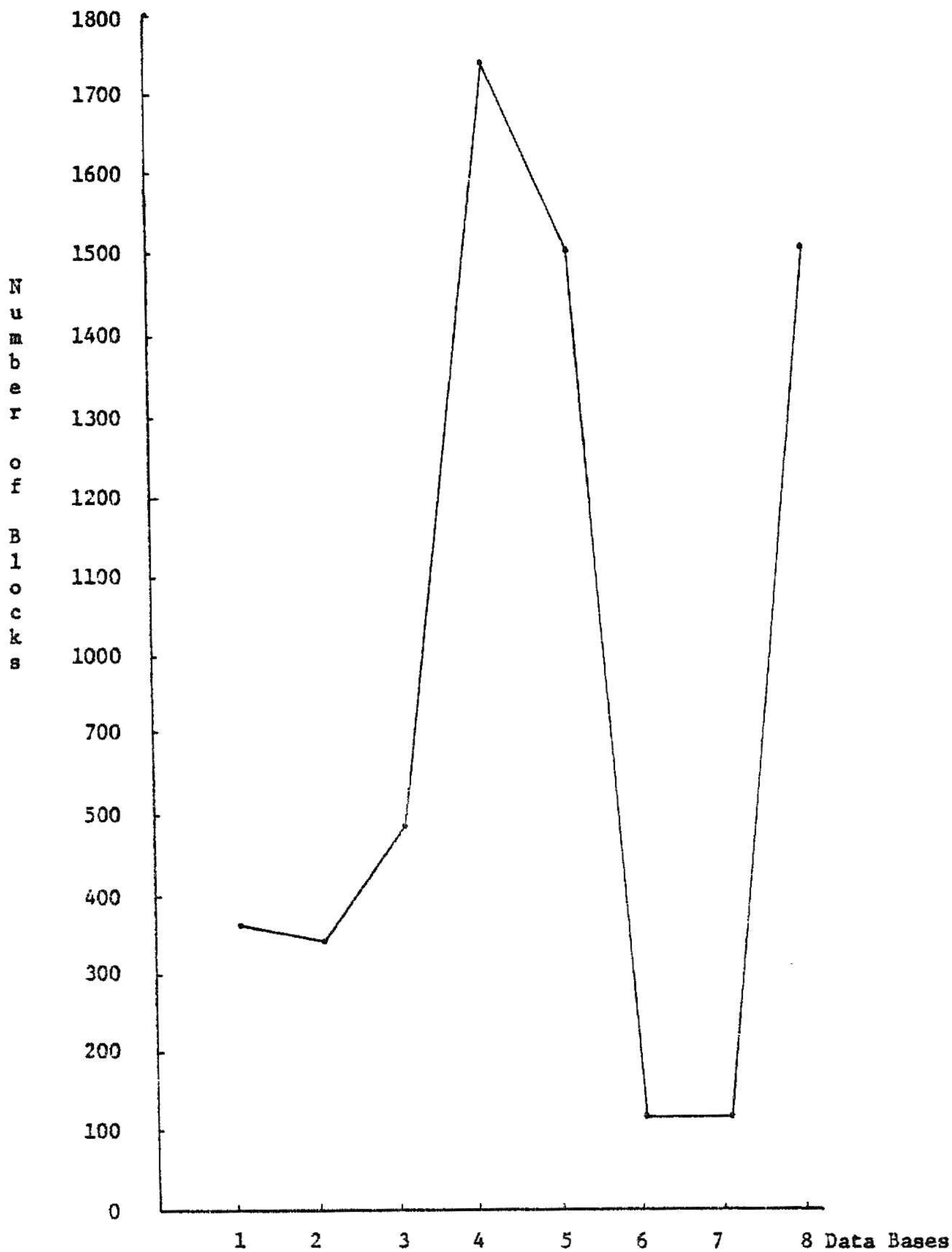

Figure 5.5

Graphic Representation of the Total Storage Requirements for "Long" Group data-base requires less storage than data-base 1. Data-base 3 presents an increment in its storage requirements since it is a larger data-base. Data-base 4 requires the higher number of blocks in the group. The reason for this is that it is the largest data-base in the group. Data-base 5 and 8 are similar in characteristics; therefore, their storage requirements are the same. Since data-base 6 and 7 are the smallest in the group both of them utilize the minimum storage requirements.

5.2 Analysis of Access-time

Access-times were obtained for the five groups of data-bases treated in this study, those access-times embrace the four query complexities described by equations 3.19 through 3.22 in chapter 3. In this section we are going to analyze only the access-times obtained for queries of complexity 4 at different levels of concurrency, we chose only this complexity because it is the most common kind of query complexity made by users in the DBMS environment. Access-times for complexities 1, 2 and 3 are given in Appendix C.

5.2.1 Levels of Concurrency

In our study we emulate the research carried out by Slonim (8) by using the same concurrency levels that he used, those levels are as follows:

A) First Level: This level utilizes 8 processors, all of them are dedicated to the I/O, since I/O is the bottleneck in most information systems, the function of those processors is to alleviate that problem. Those processors are indicated by "P" in equations 3.19 through 3.22, this first level of concurrency affects only the access-time 1 in the four query complexities.

B) Second Level: Here the number of concurrent processors is increased. Besides the 8 processors working in the I/O, 4 processors are added, one for each atomic condition per item-condition. Those processors are indicated by "CACI" in equations 3.20 through 3.22. This second level of concurrency affects only the access-time 2, 3 and 4 of complexities 2, 3 and 4.

C) Third Level: Here the level of concurrency is increased again by adding 20 processors more, each of which is dedicated to an item-condition per record-condition. Those processors are denoted by "CICR" in equations 3.21 and 3.22. This third level of concurrency affects only access-time 3 and 4 of complexities 3 and 4.

D) Fourth Level: Here the level of concurrency is increased by adding 40 processors more, each of which is devoted to a record-condition per query condition. Those processors are denoted by "CRCQ" in equation 3.22. This fourth level of concurrency affects only access-time 4 of complexity 4.

Table 5.6 shows the number of processors required at each level for each complexity (access-time). In all cases, the maximum number of processors is introduced at each level.

Level \ Complexity	1	2	3	4
1	8	8	8	8
2	8	12	12	12
3	8	12	32	32
4	8	12	32	72

Table 5.6

Number of Processors utilized at each Level

In the following section will be described the tables, graphics and analysis for the complexity 4 at differents levels of concurrency, for the five groups of data bases utilized in this study. The analysis will consist of a comparison between the results obtained for centralized processing against concurrent processing. All the improvements shown are relative to the prior level of concurrency.

5.2.2 Access-times for "Normal" Group

A) First Concurrency Level: (see tables 5.7 and 5.8 and figure 5.6) The improvements reached by concurrent over sequential processing were only of importance for data-bases 4, 5 and 8 which are the large data-bases of the group, less significant improvements were obtained for the smaller data-bases. The access-times for data-bases 5 and 8 were reduced by 100 percent and more; for the smaller data-bases, reductions were between 10 and 18 percent. The introduction of the 8 processors for handling I/O operations in the concurrent processing had little effect. All response times remained in the "poor" category.

B) Second Concurrency Level: (see tables 5.7 and 5.8 and figure 5.6) At this level the improvements in response times were very considerable for all data-bases. The greatest improvements were registered in the smaller data-bases, their times were reduced in more than 100 percent. Data-bases 5 and 8 still remain in the "poor" category while all the other fell in the "average" category.

C) Third Concurrency Level: (see tables 5.7 and 5.8 and figure 5.6) When 20 processors more are added to deal with atomic conditions the access-time is reduced tremendously in all data-bases until reach category of "good" for data-bases 1, 2, 3, 6 and 7.

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	30.030	27.055	11.388	3.240	0.955
2	30.112	27.137	11.410	3.246	0.952
3	31.812	27.350	11.622	3.458	1.165
4	43.022	28.759	13.032	4.861	2.567
5	66.512	31.687	15.960	7.796	5.502
6	32.410	27.424	11.697	3.533	1.239
7	30.012	27.125	11.397	3.233	0.940
8	95.812	35.350	19.622	11.458	9.165

TABLE 5.7

Access Time/Sec. for Query Complexity Four
Centralized and Concurrent Processing
"Normal" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	10.99	137.57	251.48	239.26
2	0	10.96	137.83	251.50	240.96
3	0	16.31	135.32	236.09	196.82
4	0	49.59	120.67	168.09	89.36
5	0	109.90	98.54	104.72	41.69
6	0	18.18	134.45	231.07	185.14
7	0	10.64	138.00	252.52	243.93
8	0	171.03	30.15	71.25	25.01

TABLE 5.8

% of Improvement for Access Time Concurrent against Centralized

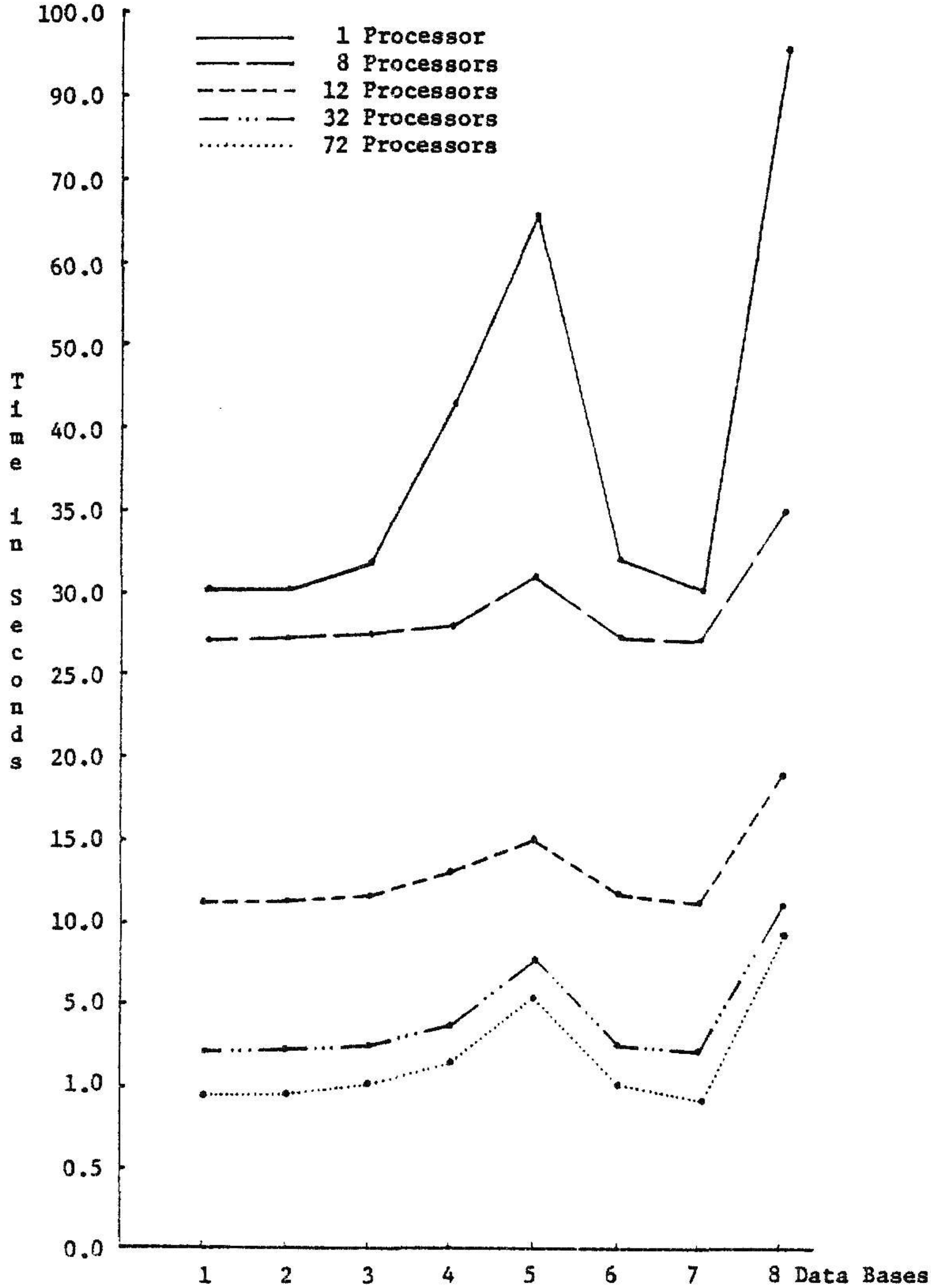

Figure 5.6

Access Time for Query Complexity 4 with 1,8,12,32 and 72 Processors "Normal" Group

Data-base 8 the only data-base with an improvement of less than 100 percent, data-bases 4 and 5 have an improvement of 168 and 104 respectively. Data-bases 4, 5 and 8 still continue in the "average" category.

D) Fourth Concurrency Level: (see tables 5.7 and 5.8 and figure 5.6) This is the highest level of concurrency (72 processors), the reduction in access-times still remain high for smaller data-bases (more than 200 percent of improvement), data-bases 4, 5 and 8 have hardly improvements of 89, 41 and 25 percent respectively. The access-times fell into the category of "excellent" for data-bases 1, 2 and 7; "very-good" for data-bases 3 and 6; "good" for data-base 4 and "average" for data-bases 5 and 8. The improvement for each level it is obtained with respect to its prior level.

5.2.3 Access-time for "Small" Group

A) First Concurrency Level: (see tables 5.9 and 5.10 and figure 5.7) Very little improvements are recorded for all data-bases at this level of concurrency, the highest improvement is recorded for data-bases 6 and 8 (11 percent) and the lowest for data-bases 1, 2, 3, and 4 (2 percent). Data-bases 5 and 7 had an improvement of 6 percent. At this level the effect of concurrency is not very effective, all the times remain as in the centralized processing in the "poor" category. That means that the response time for small data-bases is hardly improved when they are concurrently working with 8 microprocessors.

B) Second Concurrency Level: (see tables 5.9 and 5.10 and figure 5.7) The improvements recorded at this level are very significant in all the data-bases. The reduction reached in their access-

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	27.430	26.730	11.063	2.915	0.630
2	27.512	26.812	11.085	2.921	0.627
3	27.512	26.812	11.085	2.921	0.627
4	27.512	26.822	11.094	2.923	0.630
5	28.712	26.962	11.235	3.071	0.777
6	30.211	27.150	11.422	3.258	0.965
7	28.712	26.962	11.235	3.071	0.777
8	30.212	27.150	11.422	3.258	0.965

TABLE 5.9

Access Time/Sec. for Query Complexity Four
Centralized and Concurrent Processing
"Small" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	2.61	141.61	279.51	362.69
2	0	2.61	141.87	279.49	365.86
3	0	2.61	141.87	279.49	365.86
4	0	2.57	141.77	279.54	363.96
5	0	6.49	139.98	265.84	295.23
6	0	11.27	137.69	250.58	237.61
7	0	6.49	139.98	265.84	295.23
8	0	11.27	137.69	250.58	237.61

TABLE 5.10

% of Improvement for Access Time Concurrent against Centralized

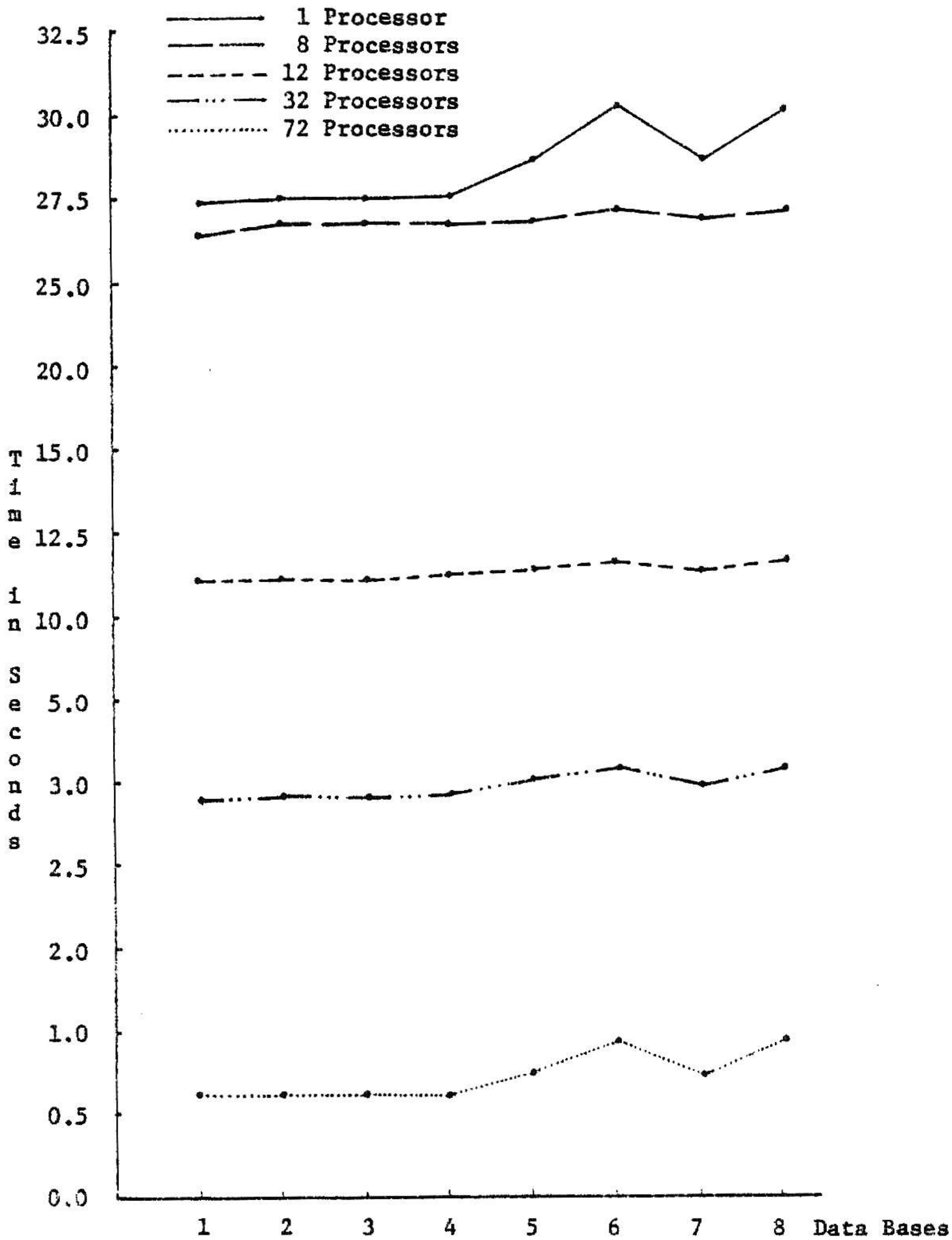

Figure 5.7

Access Times for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Small" Group

times ranges between 137 and 141 percent. Now the time for all of them fell in the category of "average".

C) Third Concurrency Level: (see tables 5.9 and 5.10 and figure 5.7) Very significant improvements were recorded again at this level, the access-times were reduced in 200 percent and more in all the cases. The times fell now in the category of "good".

D) Fourth Concurrency Level: (see tables 5.9 and 5.10 and figure 5.7) Once again the improvements recorded show a great gaining in the times reduction. Data-bases 1, 2, 3 and 4 had an improvement of more than 360 percent. Likewise data-bases 5, 6, 7 and 8 had an important improvement (237 to 295 percent) but not as high as the first four. All the times fell in the category of "excellent".

As conclusion we can say that the response-time for this group with 800 records in the 8 data-bases was hardly improved with the introduction of the first concurrency level. The second level recorded better response-times in the 8 data-bases; the improvement reached was of 250 percent and more. In the third level the improvements in response time were higher than the second level. Data-bases 1, 2, 3 and 4 were the more benefitted (279 percent in the four data-bases) this was due to the fact that they had record lengths shorter than the other data-bases. Data-bases 5 and 7 had an improvement in their response-times of 265 percent while data-bases 6 and 8 had an improvement of 250 percent. The difference in improvements is due to the fact that data-bases 6 and 8 had longer record lengths than data-bases 5 and 7. The fourth concurrency level brought improvements in response time for data-bases 1, 2, 3, 4, 5 and 7 with respect to its prior level. Data-bases 6 and 8 were less benefitted at this concurrency level, however, the improvements reached were tremendous in all cases (237 percent minimum improvement, 365 the maximum improvement).

5.2.4 Access-time for "Large" Group

A) First Concurrency Level: (see tables 5.11 and 5.12 and figure 5.8) The improvements recorded for this level are tremendously high in comparison with the other levels of concurrency. Here were registered reduction of access-time of up to 350 percent which is something very important at this level since for the others groups it hardly reaches 10 percent. Data-bases 6 and 8 were the more favored, in this group they have the largest record length. Data-bases 1, 2, 3 and 4 had an improvement of 120 percent and more, meanwhile data-bases 5 and 7 had an improvement of 260 percent. All the times were in the "poor" category.

B) Second Concurrency Level: (see tables 5.11 and 5.12 and figure 5.8) Here the reduction of access-time was not as high as in the first level, the improvement ranges between 40 and 94 percent. Here, contrary to what happened in the first level, the data-bases 6 and 8 were the ones which had the lower improvement (40 percent). The better improvement was for data-bases 1, 2, 3 and 4 (93 to 94 percent). The response-time still continues in the "poor" category for all the group.

C) Third Concurrency Level: (see tables 5.11 and 5.12 and figure 5.8) Here again we found improvements of less than 100 percent which is not bad, but certainly not as good as the third level in the other groups. The higher improvements were again for data-bases 1, 2, 3 and 4 (90 to 97 percent) as in second level. In second term

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	73.025	32.430	16.762	8.615	6.330
2	73.112	32.512	16.785	8.621	6.327
3	76.312	32.912	17.185	9.021	6.727
4	71.522	32.322	16.594	8.423	6.130
5	154.212	42.650	26.922	18.758	16.465
6	248.430	54.427	38.699	30.535	28.242
7	154.212	42.650	26.922	18.758	16.465
8	248.512	54.437	38.710	30.546	28.252

TABLE 5.11

Access Time/Sec. for Query Complexity Four
Centralized and Concurrent Processing
"Large" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	125.17	93.47	94.56	36.09
2	0	124.87	93.69	94.69	36.25
3	0	131.86	91.51	90.49	34.10
4	0	121.27	94.78	97.00	37.40
5	0	261.57	58.42	43.52	13.92
6	0	356.44	40.64	26.73	8.11
7	0	261.57	58.42	43.52	13.92
8	0	356.51	40.62	26.72	8.11

TABLE 5.12

% of Improvement for Access Time Concurrent against Centralized

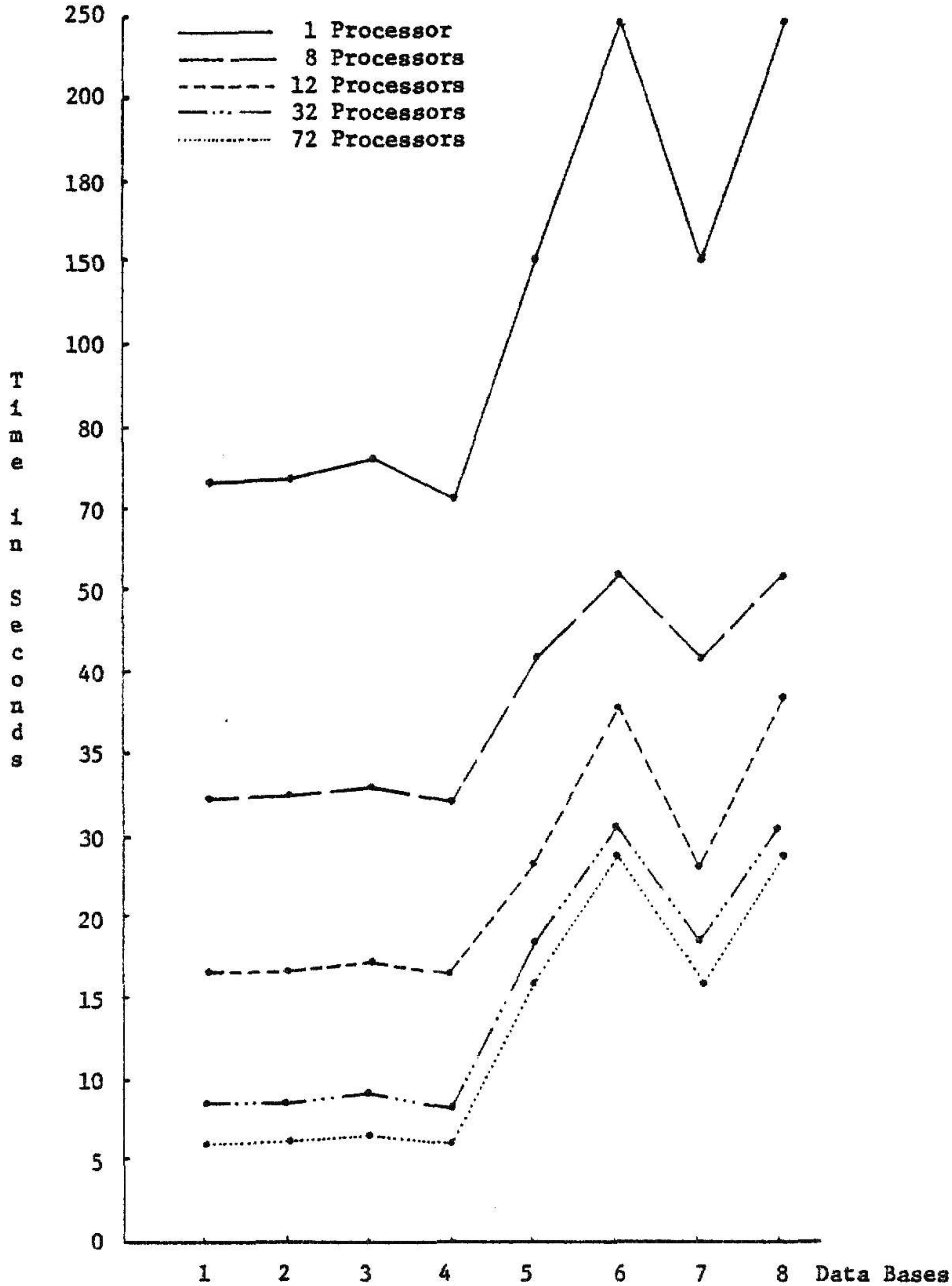

Figure 5.8

Access Times for Query Complexity 4 with 1,8,12,32 and 72 Processors
"Large" Group

were the improvements for data-bases 5 and 7 (43 percent) and in the last term data-bases 6 and 8 only won 26 percent of improvement. The times fell in the following categories: data-bases 1, 2, 3 and 4 "average" all the others in the "poor".

D) Fourth Concurrency Level: (see tables 5.11 and 5.12 and figure 5.8) Contrary to what happened in others groups at this level of concurrency, this group recorded a low improvement in the 8 data-bases. The higher as in the prior level of concurrency was recorded for data-bases 1, 2, 3 and 4 (34 to 38 percent). Data-bases 5 and 7 again recorded the same improvement (13 percent). Data-bases 6 and 8 once again recorded the lower improvement (8 percent). The times for data-bases 1, 2, 3 and 4 fell in the "average" category, all the other remain in the "poor" category.

In conclusion the processors dedicated to the I/O were of great importance. When 8 processors were used to deal with I/O operations the highest improvements in response time were recorded. When 4 processors more were added to deal with atomic-conditions the improvements were not as dramatic as in the first level, nevertheless the access-times were reduced in almost one half. At the point where 20 processors more were dedicated to deal with item-conditions the improvement in response time was not very good in all the cases. Data-bases 1, 2, 3 and 4 hardly reach a disminishing of one half, all the others had little improvement. At fourth concurrency level when 40 processors more are dedicated to deal with record-conditions the improvement reaches the lower level, which means that large data-bases do not have high improvements even at high levels of concurrency. Data-bases with same number of records and same record lengths gave always same access-times, this is the case of data-bases 5 and 7 and 6 and 8. Finally, we can say that large data-bases with long record lengths had the poorest benefits in this group.

5.2.5 Access-times for "Short" Group

A) First Concurrency Level: (see tables 5.13 and 5.14 and figure 5.9) At this level the effects recorded by the use of concurrency were not very high. The highest improvement (10 to 12 percent) was reached by the large data-bases 4, 5 and 8 and the lowest (0.97 percent) for small data-bases 6 and 7 the other three data-bases had an improvement of 2 to 3 percent. All the times remain in the "poor" category.

B) Second Concurrency Level: (see tables 5.13 and 5.14 and figure 5.9) Here the improvement was tremendous for each data-base. Better improvements were recorded for small and medium data-bases (141 to 142 percent), larger data-bases recorded 136 to 138 percent. All the response times dropped into the "average" category.

C) Third Concurrency Level: (see tables 5.13 and 5.14 and figure 5.9) This level showed great improvements, 246 percent and more for each data-base. All the response times dropped into the "good" category.

D) Fourth Concurrency Level: (see tables 5.13 and 5.14 and figure 5.9) At this level very high improvements were recorded for the small data-bases 6 and 7 (405 percent). Data-base 4 the largest in the group, recorded the lower improvement (225 percent). Medium size data-bases recorded an important improvement of 344 to 365 percent. The response time for data-base 4 belongs now to the category of "very good", all the others belong to the "excellent" cate-

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	27.430	26.730	11.063	2.915	0.630
2	27.512	26.812	11.085	2.921	0.627
3	27.812	26.850	11.122	2.958	0.665
4	30.622	27.209	11.482	3.311	1.017
5	30.012	27.125	11.397	3.233	0.940
6	27.012	26.750	11.022	2.858	0.565
7	27.012	26.750	11.022	2.858	0.565
8	30.012	27.125	11.397	3.233	0.940

TABLE 5.13

Access Time/Sec. for Query Complexity Four
Centralized and Concurrent Processing
"Short" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	2.61	141.61	279.51	362.69
2	0	2.61	141.87	279.49	365.86
3	0	3.58	141.41	275.99	344.81
4	0	12.54	136.97	246.78	225.56
5	0	10.64	138.00	252.52	243.93
6	0	0.97	142.69	285.65	405.84
7	0	0.97	142.69	285.65	405.84
8	0	10.64	138.00	252.52	243.93

TABLE 5.14

% of Improvements for Access Time Concurrent against Centralized

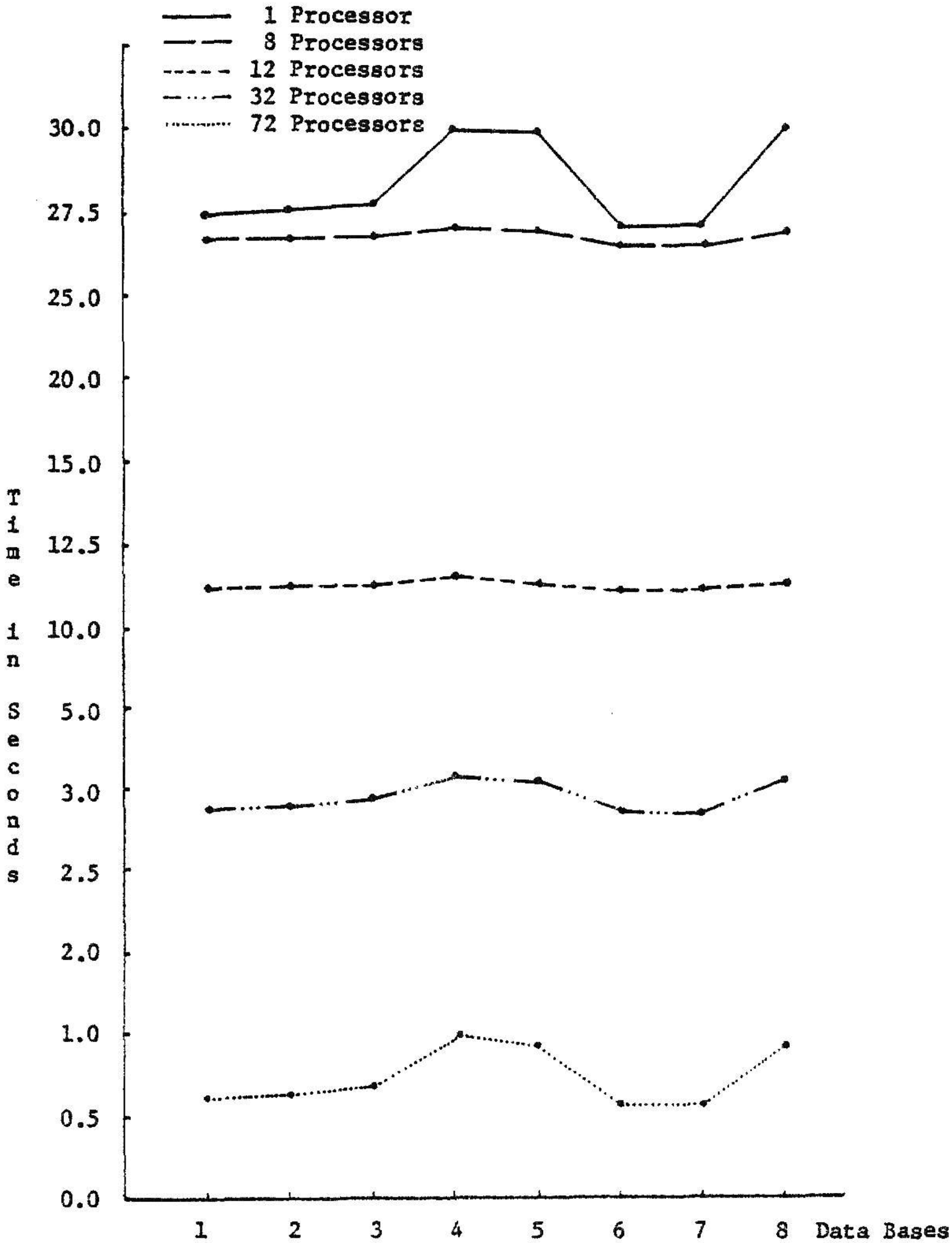

Figure 5.9

Access Times for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Short" Group

gory.

As conclusion for this group we can say that data-bases ranging from small (1296 records) to large (18573 records) having as special characteristic the same record length (20 characters) will gain very importants improvements when they work at levels 2, 3 and 4. The first level of concurrency recorded very little improvements (less than 1 percent) for the small data-bases 6 and 7; better improvement (2 to 3 percent) was recorded for medium data-bases 1, 2 and 3 and much better but still too low improvements (10 to 12 percent) for large data-bases 4, 5 and 8, as the reader can see, when were introduced 8 processors to deal with I/O functions the results were not satisfactory. At the second, third and fourth level the improvements were high, which means that the functions performed by the processors dedicated to deal with atomic-conditions, item-conditions and record-conditions had an important role at their respective levels, when those levels were executed the data-bases which had the higher improvement were small data-bases which is contrary to what happened at level 1.

5.2.6 Access-times for "Long" Group

A) First Concurrency Level: (see tables 5.15 and 5.16 and figure 5.10) The improvements in response time recorded at this level were very important for the large data-bases 4, 5 and 8 (266 to 282 percent). Small data-bases 6 and 7 reached only 32 to 35 percent and the medium size data-bases 1, 2 and 3 recorded from 59 to 124 percent respectively. All the response times were in the category of "poor".

B) Second Concurrency Level: (see tables 5.15 and 5.16 and

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	46.580	29.124	13.456	5.309	3.024
2	59.245	30.779	15.051	6.887	4.594
3	73.163	32.518	16.791	8.627	6.333
4	157.853	43.113	27.385	19.215	16.921
5	171.204	44.774	29.046	20.882	18.589
6	38.245	28.154	12.426	4.262	1.969
7	38.245	28.154	12.426	4.262	1.969
8	171.204	44.774	29.046	20.882	18.589

TABLE 5.15

Access Time/Sec. for Query Complexity Four
Centralized and Concurrent Processing
"Long" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	59.93	116.43	153.45	75.56
2	0	92.48	104.49	118.54	49.91
3	0	124.99	93.66	94.63	36.22
4	0	266.13	57.43	42.51	13.55
5	0	282.37	54.14	39.09	12.33
6	0	35.84	126.57	191.55	116.45
7	0	35.84	126.57	191.55	116.45
8	0	282.37	54.14	39.09	12.33

TABLE 5.16

% of Improvement for Access Time Concurrent against Centralized

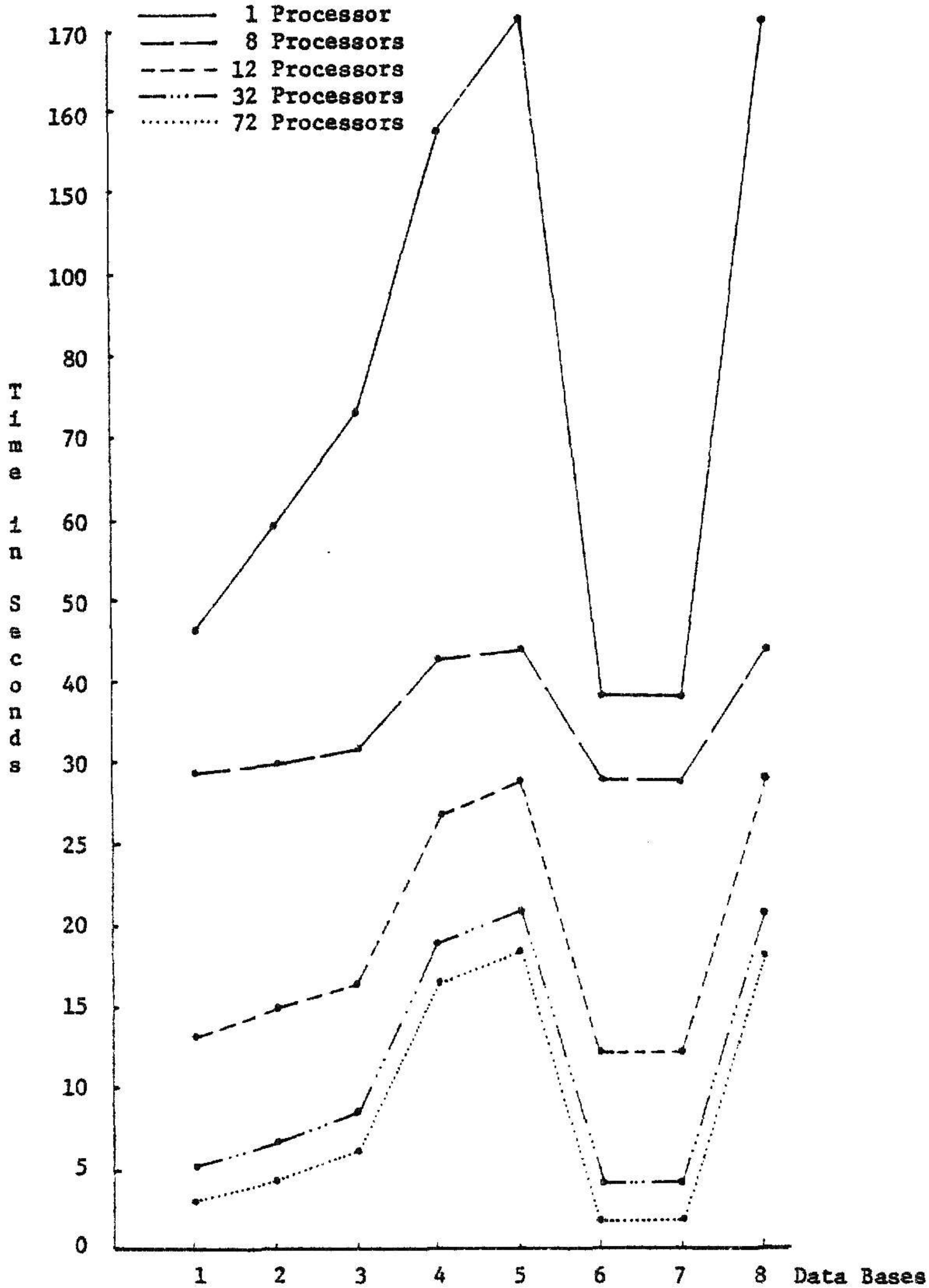

Figure 5.10

Access Times for Query Complexity 4 with 1,8,12,32 and 72 Processors
"Long" Group

figure 5.10) The improvements recorded here were important for small data-bases 6 and 7 (126 percent) and medium data-bases 1, 2 and 3 with 116, 104 and 93 percent respectively, less important for large data-bases 4, 5 and 8 (54 to 57 percent). Only response time 1, 6 and 7 fell into the "average" category all the others remain in the "poor".

C) Third Concurrency Level: (see tables 5.15 and 5.16 and figure 5.10) Data-bases 6 and 7 recorded the highest improvement (191 percent). Medium data-bases 1, 2 and 3 were reduced in their access-times by 153, 118 and 94 percent respectively. Data-bases 4, 5 and 8 recorded the lowest improvements only 42, 39 and 39 percent respectively. The response times were as follow: data-bases 1, 2, 3, 6 and 7 were in the "average" category; all the others continue in the "poor" category.

D) Fourth Concurrency Level: (see tables 5.15 and 5.16 and figure 5.10) Less important improvements were recorded at this level in all data-bases, however, small data-bases continue having a reduction in their access-times of more than 100 percent which is of great importance. Data-bases of medium size like 1, 2 and 3 reported a reduction of 75, 49 and 36 percent respectively; only the large data-bases 4, 5 and 8 did not get important improvements 13, 12 and 12 percent respectively. The response times were as follow: data-bases 6 and 7 fell into the "very good" category. Data-base 1 in the "good" category. Data-bases 2 and 3 in the "average" and rest in the "poor" category.

We can conclude that at first level of concurrency the access-times were more strongly affected in those cases where the size of the data-base was larger than the others ones. This is due to the fact that the combination of a large data-base with long record-length is well handled in its I/O functions by 8 processors. When small data-bases with long record-length are utilized the I/O functions still have some bottleneck as in centralized processing. When the second, third and fourth levels were exercised the improvements recorded were always more significant for small data-bases; medium and large data-bases had from light gains to big decrements in the improvement of the access-times.

It must be pointed out that the "Small" and "Short" groups had almost the same behavior in their access-times though they were data-bases with different characteristics. The small number of records in the "Small" group was compensated by its different record lengths and in the "Short" group the short record length was compensated by its different data-bases sizes. Graphics for this five groups are described by figures 5.6 through 5.10.

5.3 Analysis of Costs

The following analysis of costs is carried out for the five groups of data-bases utilizing centralized and concurrent processing. The cost that is mentioned in this report corresponds to the cost per query made to the data-base.

5.3.1 Costs for "Normal" Group: (see tables 5.17 and 5.18 and figure 5.11) The cost recorded for concurrent processing is always higher than for centralized processing. At first level of concurrency the gains were higher for large data-bases 5 and 8, this can be explained simply: the cost for eight extra processors is smaller than the cost of running a large centralized machine. Costs for

CONC. LEVEL / DATA BASE	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	4.29	3.66	1.03	0.11	0.01
2	4.27	3.67	1.03	0.11	0.01
3	4.58	3.71	1.05	0.11	0.01
4	6.83	3.94	1.20	0.17	0.04
5	12.19	4.42	1.50	0.29	0.10
6	4.65	3.71	1.06	0.12	0.01
7	4.23	3.67	1.03	0.11	0.01
8	20.37	5.03	1.91	0.47	0.20

TABLE 5.17

Cost/$ for Query Complexity Four Centralized and Concurrent Processing "Normal" Group

CONC. LEVEL / DATA BASE	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	17.2	255.3	836.3	1000.0
2	0	16.3	256.3	836.3	1000.0
3	0	23.4	253.3	854.5	1000.0
4	0	73.3	228.3	605.8	325.0
5	0	175.7	194.6	417.2	190.0
6	0	25.3	250.0	783.3	1100.0
7	0	15.2	256.3	836.3	1000.0
8	0	304.9	163.3	306.3	135.0

TABLE 5.18

% of Improvement for Cost Concurrent against Centralized

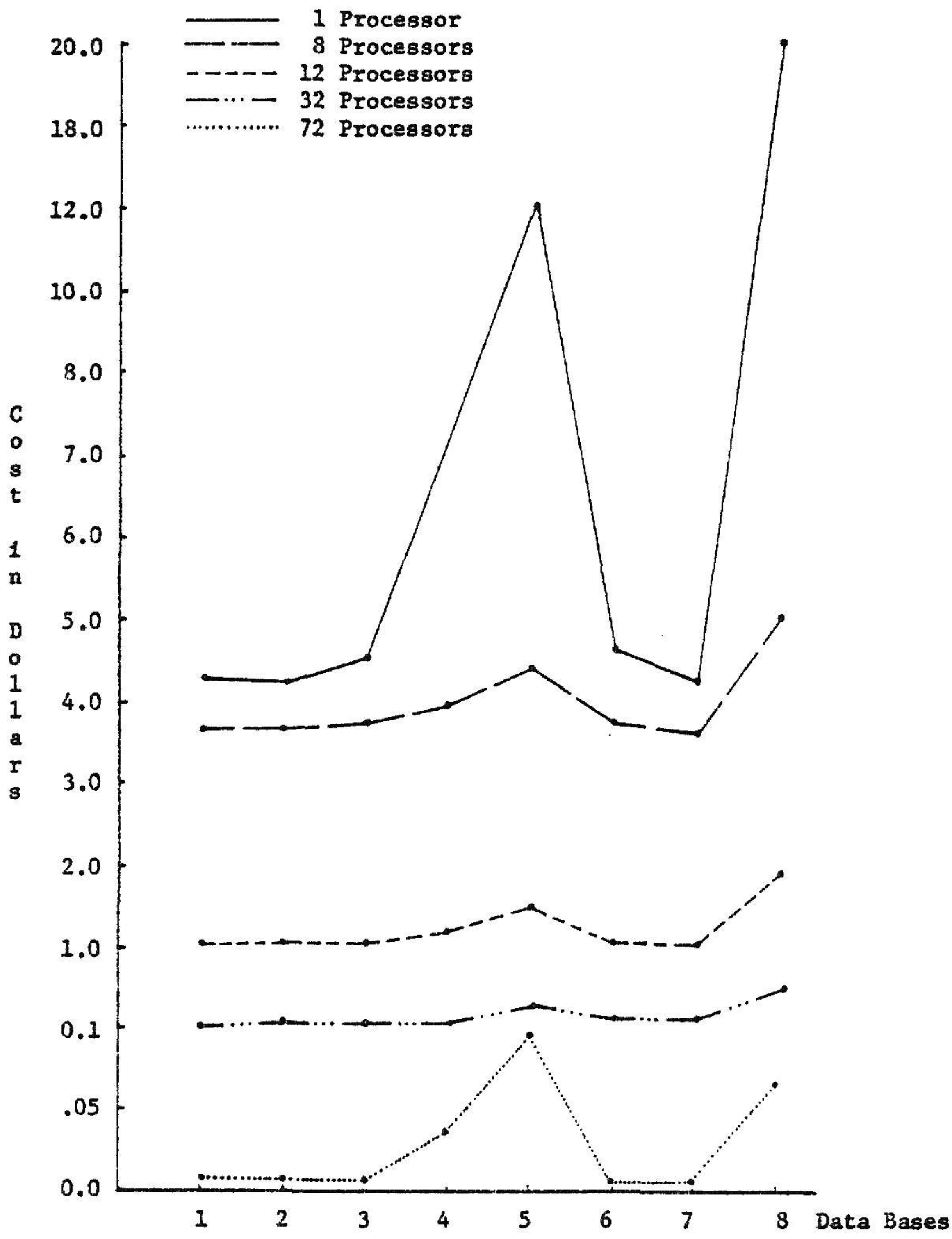

Figure 5.11

Cost for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Normal" Group

centralized processing ranges from $4.23 to $20.37 whereas for centralized at first concurrency level ranges from $3.66 to $5.03. When twelve processors were used the cost was reduced in all the cases and now the cost ranges from $1.03 to $1.91. Large data-bases had higher costs than small data-bases, this is due to the fact that large data-bases require more storage-space. At third level the cost for concurrent processing was reduced tremendously; all the costs range between $0.11 and $0.47. Better cost was recorded for small data-bases since they require less storage. When the fourth level was utilized the cost-benefit showed a very important improvement for data-bases with small storage-requirements. Likewise large data-bases had a significant decrement in their costs; the improvement with relation to the third level was from 135 to 325 percent.

5.3.2 Costs for "Small" Group: (see tables 5.19 and 5.20 and figure 5.12) The cost for running this group of data-bases in a large centralized computer ranges from $3.81 to $4.27, the same group, when was runned with 8 processors dedicated to deal with I/O functions recorded low improvements in all the cases from 5.2 to 16.3 percent; which mean a reduction in the cost from 19 to 60 cents. At the second concurrency level better improvements were recorded in relation to the first level they ranged from 256 to 265 percent, the four processors added started to realize the benefits of concurrency, the cost was now little lower for data-bases with short record length. The third level brought very high decrements in the cost for running the group; the improvements were between 800 and 1000 percent which constitutes a fabulous saving when processing is carried out at this level. When the group was running at the full concurrency level

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	3.82	3.61	0.99	0.09	0.009
2	3.81	3.62	0.99	0.09	0.009
3	3.81	3.62	0.99	0.09	0.009
4	3.81	3.62	1.00	0.09	0.009
5	4.01	3.64	1.00	0.10	0.011
6	4.26	3.67	1.03	0.11	0.014
7	4.01	3.64	1.01	0.10	0.011
8	4.27	3.67	1.03	0.11	0.014

TABLE 5.19

Cost/$ for Query Complexity Four Centralized and Concurrent Processing "Small" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	5.8	264.6	1000.0	900.0
2	0	5.2	265.6	1000.0	900.0
3	0	5.2	265.6	1000.0	900.0
4	0	5.2	262.0	1011.1	900.0
5	0	10.1	264.0	900.0	809.0
6	0	16.0	256.3	836.3	685.7
7	0	10.1	260.3	910.0	809.0
8	0	16.3	256.3	836.3	685.7

TABLE 5.20

% of Improvement for Cost Concurrent against Centralized

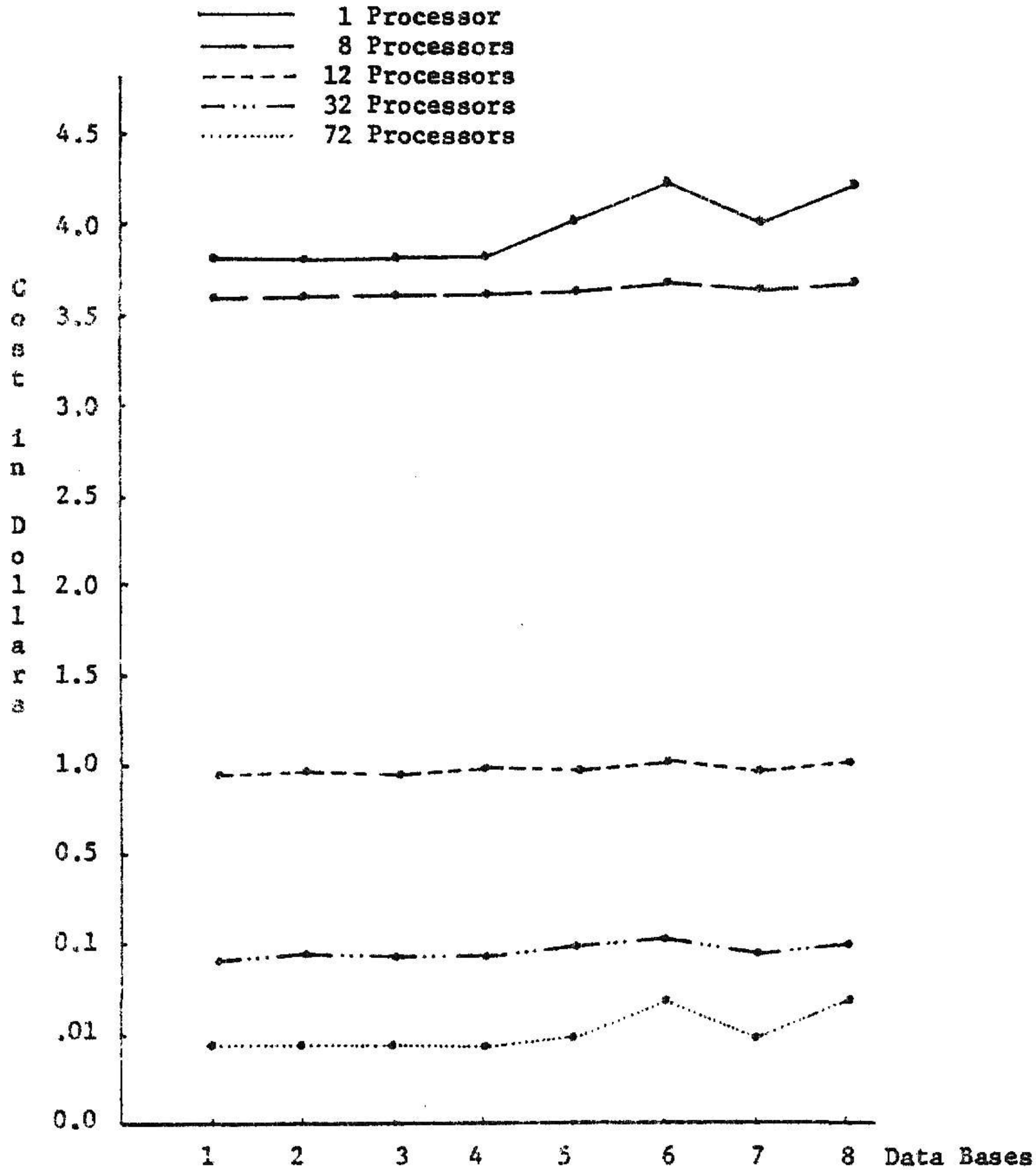

Figure 5.12

Cost for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Small" Group

the improvements reached were also very high for each case but lower than the third level. Data-bases 6 and 8 which were the ones with longer record length, they recorded the minimum improvements in their costs, contrary to this short and medium record lengths shown better improvements.

5.3.3 Costs for "Large" Group: (see tables 5.21 and 5.22 and figure 5.13) This group recorded the highest cost for centralized processing. At the first level we found that large data-bases 5, 6, 7 and 8 with long record length recorded the highest improvements among the four concurrency levels, this mean that data-bases with similar characteristics to those of the data-base mentioned will have much better cost-benefit when 8 processors are dedicated to the I/O functions. Costs were dropped from $13.96 to $94.02 for centralized processing to $4.55 to $8.66 for this first level. The improvement recorded at the second concurrency level was much lower for data-bases with long record length the explanation to this is because the bigger-storage-requirements for those data-bases. The third concurrency level showed the second better improvement among the levels, here the gains in the cost were higher for data-bases with short record length. The cost was dropped from $1.61 to $4.41 for twelve processors to $0.33 to $1.76 for thirty two processors. The improvements in cost-benefit at fourth level of concurrency were not too high although still of importance; data-bases with long record length as in second and third level were the ones with less improvement the explanation is the same given above, greater storage-requirements, this level of concurrency dropped the costs for this group in moderate fashion ranging from $0.12 to $0.14 for data-bases 1, 2, 3 and 4 (which have

CONC. LEVEL / DATA BASE	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	14.87	4.59	1.61	0.34	0.13
2	14.37	4.58	1.61	0.34	0.13
3	15.25	4.64	1.65	0.36	0.14
4	13.96	4.55	1.58	0.33	0.12
5	43.43	6.38	2.81	0.90	0.48
6	92.43	8.62	4.39	1.74	1.08
7	42.83	6.35	2.80	0.89	0.47
8	94.02	8.66	4.41	1.76	1.10

TABLE 5.21

Cost/$ for Query Complexity Four Centralized and Concurrent Processing "Large" Group

CONC. LEVEL / DATA BASE	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	223.9	185.0	373.5	161.5
2	0	213.7	184.4	373.5	161.5
3	0	228.6	181.2	358.3	157.1
4	0	206.8	187.9	378.7	175.0
5	0	580.7	127.0	212.2	87.5
6	0	972.2	96.3	152.2	61.1
7	0	574.4	126.7	214.6	89.3
8	0	985.6	96.3	150.5	60.0

TABLE 5.22

% of Improvement for Cost Concurrent against Centralized

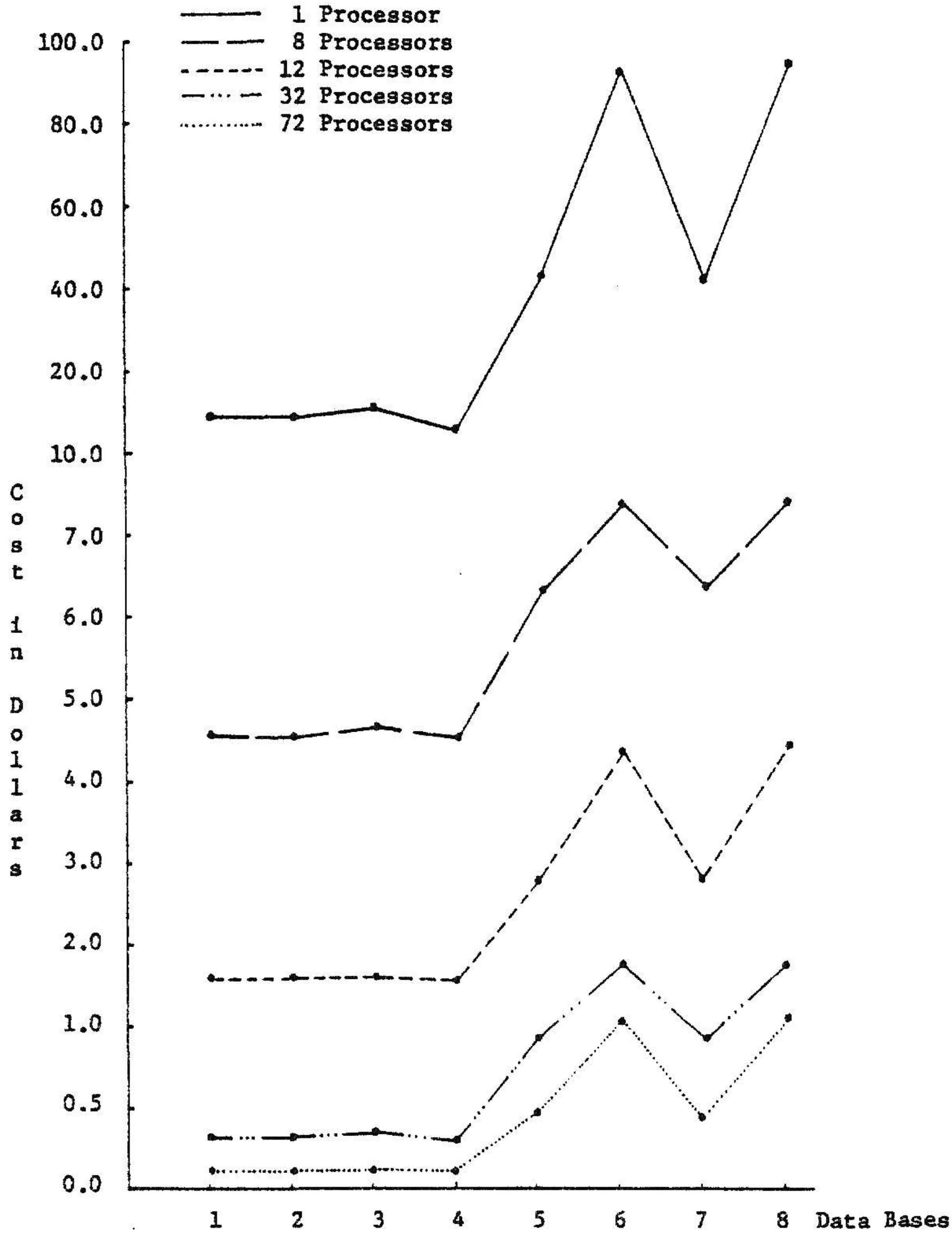

Figure 5.13

Cost for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Large" Group

short record length) to $0.47 to $1.10 for data-bases 5, 6, 7 and 8 with long record length.

5.3.4 Costs for "Short" Group: (see tables 5.23 and 5.24 and figure 5.14) The use of concurrency in this group did not pay great improvements at the first level. The better cost was recorded for data-base 4 which is the largest data-base, at this level small data-bases 6 and 7 had an improvement of 3.3 percent each. Medium size data-bases 1, 2, and 3 recorded 6.6, 5.8 and 7.1 percent respectively. Large data-bases had an improvement between 19.5 and 21.4. The second concurrency level had significant improvements in all the cases, they range from 254 to 265 percent. The cost dropped from $3.61 to $3.69 (first level) to $0.99 to $1.04 for this level. The reason for this is that the processors dedicated to deal with atomic conditions reduced the access-time tremendously therefore the cost was reduced. The third and fourth levels had very impresive improvements. They range from 587 to 1025 percent which brought down the cost to very low levels. The cost for processing this group at third concurrency level is as follow: $0.09 for the small data-bases 6 and 7; $0.09 to $0.10 for medium data-bases 1, 2 and 3 and $0.11 for large data-bases, whereas for the fourth level ranges from $0.008 to $0.016. After the first level the larger data-bases 4, 5 and 8 started to obtain lower cost-benefits this was due to the fact that they utilize more storage.

5.3.5 Costs for "Long" Group: (see tables 5.25 and 5.26 and figure 5.15) The first concurrency level showed up an important improvement in medium and large data-bases, between 100 and 634 percent. Data-bases of small size recorded only an improvement of 45 to 49 percent. Cost were dropped greatly at this level. At the second level the

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	3.85	3.61	0.99	0.09	0.009
2	3.83	3.62	0.99	0.09	0.009
3	3.89	3.63	1.00	0.10	0.010
4	4.48	3.69	1.04	0.11	0.016
5	4.40	3.68	1.03	0.11	0.015
6	3.73	3.61	0.99	0.09	0.008
7	3.73	3.61	0.99	0.09	0.008
8	4.40	3.68	1.03	0.11	0.015

TABLE 5.23

Cost/$ for Query Complexity Four Centralized and Concurrent Processing "Short" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	6.6	264.6	1000.0	900.0
2	0	5.8	265.6	1000.0	900.0
3	0	7.1	263.0	900.0	900.0
4	0	21.4	254.8	845.4	587.5
5	0	19.5	257.2	836.3	633.3
6	0	3.3	264.6	1000.0	1025.0
7	0	3.3	264.6	1000.0	1025.0
8	0	19.5	257.2	836.3	633.3

TABLE 5.24

% of Improvement for Cost Concurrent against Centralized

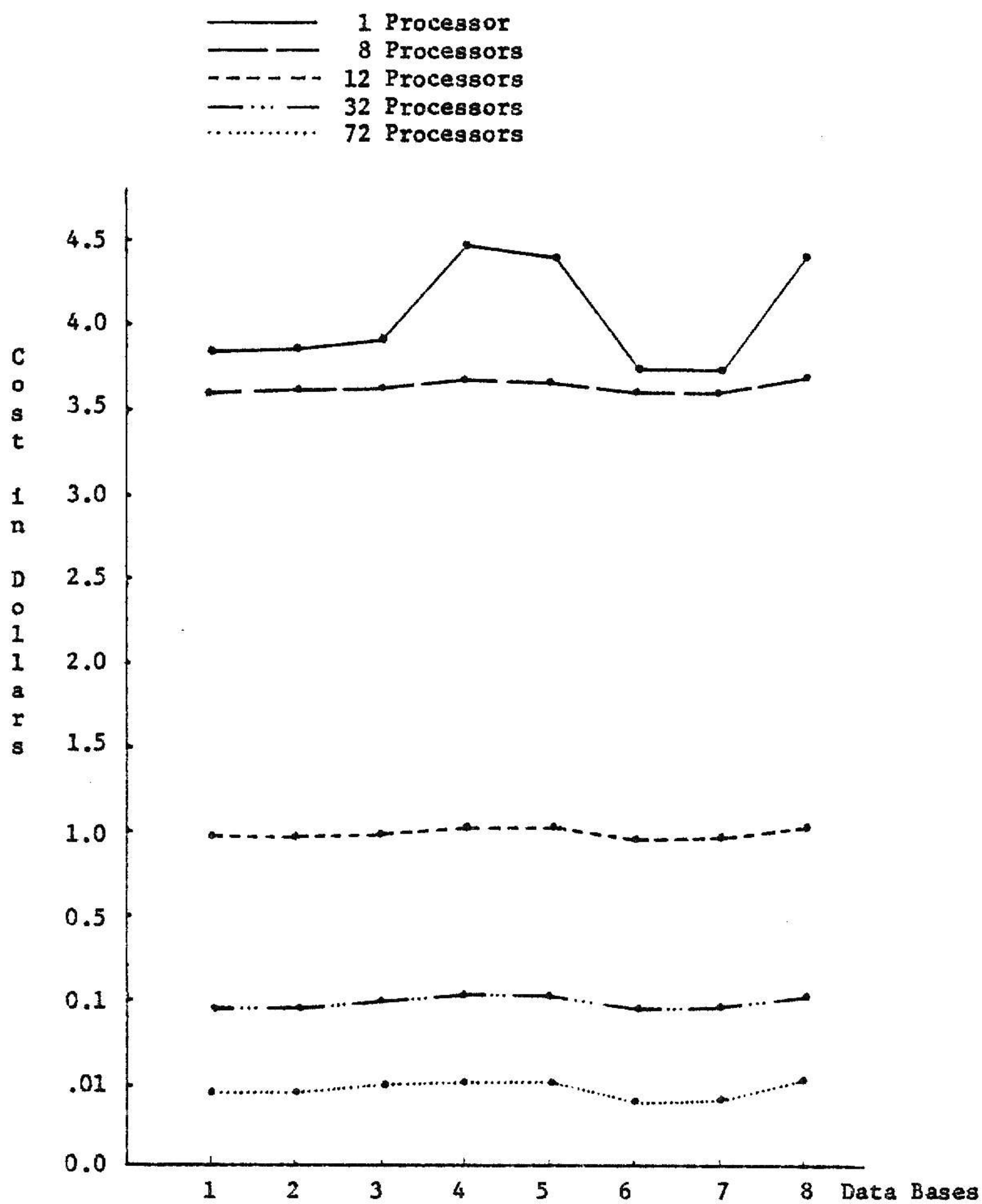

Figure 5.14

Cost for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Short" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	8.06	4.03	1.25	0.19	0.05
2	10.19	4.26	1.40	0.25	0.08
3	13.65	4.55	1.59	0.33	0.12
4	33.36	6.13	2.94	0.98	0.55
5	49.31	6.71	3.05	1.01	0.55
6	5.72	3.92	1.19	0.17	0.04
7	5.72	3.83	1.13	0.14	0.03
8	49.31	6.71	3.05	1.01	0.55

TABLE 5.25

Cost/$ for Query Complexity Four Centralized and Concurrent Processing "Long" Group

DATA BASE \ CONC. LEVEL	CENT. 1 PROCESSOR	FIRST 8 PROCESSORS	SECOND 12 PROCESSORS	THIRD 32 PROCESSORS	FOURTH 72 PROCESSORS
1	0	100.0	222.4	557.8	280.0
2	0	139.2	204.2	460.0	212.5
3	0	200.0	136.1	381.8	175.0
4	0	444.2	108.5	200.0	78.1
5	0	634.8	120.0	201.9	83.6
6	0	45.9	229.4	600.0	325.0
7	0	49.3	238.9	707.1	366.6
8	0	634.8	120.0	201.9	83.6

TABLE 5.26

% of Improvement for Cost Concurrent against Centralized

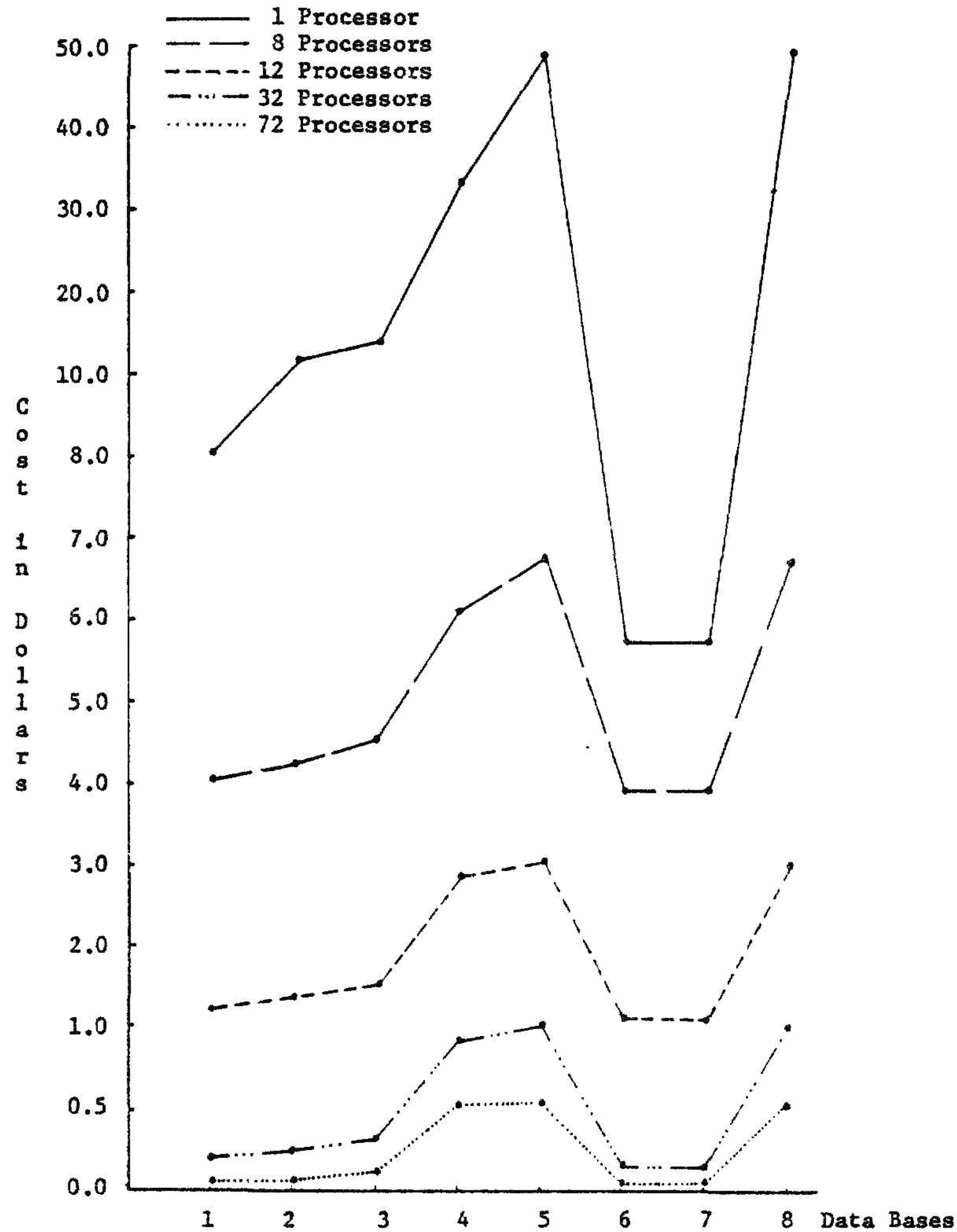

Figure 5.15

Cost for Query Complexity 4 with 1,8,12,32 and 72 Processors

"Long" Group

improvement was important too in all the cases were reported improvements between 108 and 238 percent, therefore, the cost was reduced from $3.82 to $6.71 (for first level) to $1.13 to $3.05 for this level. The small data-bases 6 and 7 were the ones with more cost-benefit since the processing of them was the cheaper of the group, this was due to the fact that they require less storage. The third level recorded the higher cost for this group. Costs were dropped tremendously for small and medium data-bases from 371 to 707 percent, data-bases with more storage-requirements had less but still important improvements (200 percent). At full level of concurrency there was a significant decrement in the cost-benefit of the group, however, the improvement was still important for data-bases of small and medium size 175 to 325 percent.

After all the analysis carried out for times and costs we conclude that concurrent processing was always faster and cheaper than centralized processing. The different characteristics of the five groups analyzed did not affect that fact. As access-times are very closely correlated with costs, the smaller the time during which it is necessary to use the CPU and secondary storage-devices, the lower the cost.

All the improvements were taken with respect to the prior level of concurrency. As can be seen times and costs were closely correlated to data-base characteristics. The longer the number of records and the longer the record length the slower the response time and the higher the cost.

VI. Conclusion

Summary.

In this study we have continued with the research initiated by Slonim (8) by using his concurrent equations for access-times and costs and recurring to the equations for centralized processing given by Cardenas (1).

In our study we made an emulation to the research carried out by Slonim (8) by comparing centralized processing against concurrent but in our case we used five groups of data-bases which were named: "Normal", "Small", "Large", "Short" and "Long". The "Normal" group has identical characteristics to the set of data-bases used by Slonim the reason of this was to have a solid basis to start with. The "Small" group has a variation in the number of records of each data-base of the group, it was set up to 800 records in the eight data-bases, remaining all the other characteristics unchanged. The "Large" group is distinguished because of its large data-bases; each data-base was augmented to 51000 records, others characteristics remain without change. The "Short" and "Long" groups have a variation only in their record lengths, using 20 and 800 characters respectively the other characteristics were unchanged.

In this study we do not analyze the equations given by Cardenas (1) and Slonim (8) they are only described and used in the same way that those researchers did. A new set of equations for access-time was developed for concurrent processing. This group of equations (Eq. 3.19 through 3.22) are based on the equations given by Slonim (8).

These new equations provide a group of equations more closely related to the equations for centralized processing given by Cardenas (1). The first part of this study was to verify the times and costs recorded in the Slonim's research for concurrent processing and also to determine the times recorded in the Cardenas' research for centralized processing. The second part was the experiment, here we used the five groups of data-bases described above. Each group was tested using centralized and concurrent processing at four levels of concurrency. There were four concurrency levels. First concurrency level: here were assigned 8 microprocessors to alleviate the bottleneck produced by I/O operations in an information system. Second concurrency level: here were added 4 microprocessors more to deal with atomic-conditions per item-condition. Third concurrency level: here were added 20 microprocessors more dedicated to deal with item-conditions per record-condition. Fourth concurrency level: here were added 40 microprocessors more dedicated to deal with record-conditions per query-conditions. For the experiments with centralized processing it is assumed the use of a large centralized machine.

The experiments carried out in this study reported four access-times for centralized and concurrent processing, the last mentioned at each level of concurrency; those access-times correspond to each kind of query complexity. For centralized processing the query complexity is given by equations 3.15 through 3.18 and for concurrent processing the query complexity is given by equations 3.19 through 3.22. The first level of query complexity is when the user does not use any qualification in the query done to the data-base. The second level of complexity is when the user is allowed to use besides to logical operators the relational operator OR in the queries made to the data-base. The third query complexity permits the use of logical operators and the relational operator AND. In fourth query complexity level the user is allowed to use logical operators and the relational operators AND and OR in his query. In this study all the tables and figures shown in Chapter V are referred only to the fourth kind of query complexity since we considered that this is the most usual query in the DBMS environment. Improvements in access-times were calculated for the fourth query complexity at all levels of concurrency for the five groups of data-bases, those improvements are given in Chapter V. Tables for the other three query complexities at four levels of concurrency for the five groups under study are given in Appendix C.

Another part of this study was dedicated to costs. In this area we utilized the equations given by Slonim (8) for the computation of costs incurred in the processing of centralized and distributed architectures. The costs were calculated for the four query complexities at all levels of concurrency for the five groups of data-bases. We chose to analyze only complexity four at all concurrency levels the reasons of this are the same given for access-times. The improvement in cost was calculated for the fourth complexity at all levels, the improvement at any specific concurrency level is calculated to its prior level.

Analysis.

The results recorded in this study lead us to conclude that concurrency at any level is a powerful alternative for those who perform in centralized environments. Cost and response-time depend on data-base characteristics. The user must balance the data-base characteristics and the level of concurrency in order to obtain the maximum cost-benefit and the minimum response-time. In our study we have seen that a higher concurrency level does not always record the best improvements in time or cost with respect to its prior level of concurrency, however, higher concurrency levels recorded always lower times therefore lower costs. Since times and costs are closely correlated, the smaller the time during which it is necessary to hold the CPU and secondary storage-devices, the lower the cost.

The system's architecture affects greatly the throughput, any system in order to achieve maximum throughput must minimize the response-time, in our study, this, was always obtained at full concurrency level. That is, if the system has the maximum number of available processors working concurrently then will be recorded the maximum throughput. In this study we have seen how the access-time of any data-base not only depends on a single factor such as size but also on a related group of factors such as record length and architecture of the system. Centralized processing invariably leads to slow access-times and high costs, concurrent processing always recorded better access-times and lower costs.

Future Work.

Further study of the effects of various characteristics of data-bases on concurrent processing is required.

The study carried out in Chapter V may be repeated using different test cases, varying other characteristics of the data-bases in order to see how access-time and cost are affected for those new characteristics.

In this study we utilized only data-bases with a single structure (i.e., inverted file) but the study could be repeated for unstructured data-bases like relational data-bases.

For this study it was assumed that the system consisted of homogeneous software and hardware; this stimulates interest in how access-time and cost will be affected using heterogeneous hardware and software?

This study is only concerned with access-times and costs for retrieval from a data-base; further research can also study the updating of a data-base.

VII. Bibliography

1. A.F. Cardenas, "Analysis and Performance of Inverted Data Base Structures", CACM, Vol. 18, Num. 5 (May 1975), pp.253-263.

2. A.F. Cardenas, Data Base Management Systems, Allyn and Bacon, Inc., 1979.

3. K.M. Chandy, "Models of Distributed Systems", Proceeding of Third International Conference on V.L.D.B. (October 6-8, 1977), pp.105-120.

4. C.W. Bachman, "Trends in Database Management 1975", AFIPS Conference Proceedings, 36 (1970), pp. 313-322.

5. C.J. Date, An Introduction to Data Base System, (Second Edition) Addison-Wesley Publishing Company (1977).

6. Kansas State University "ITEL AS5 Cost" Kansas State University Computer Center, Manhattan, Kansas 66502 (May 1979).

7. F.J. Maryanski et al., "Distributed Data Base Management Using Minicomputers", Infotech State-of-the-Art Report, "Mini's Versus Mainframes (1978).

8. J. Slonim, "Generalized Distributed Information Management Architecture" Doctoral Dissertation, Kansas State University, Manhattan, Kansas, March 1979.

9. B. Sundgren, Theory of Data Base, Petrocelli/Charter (1975).

10. R.H. Thomas, "A Resource Sharing Executive for the ARPANET", Proceedings AFIPS NCC, AFFIPS PRESS, Vol. 42 (1973), pp.155-163.

APPENDIX A

Parameters and Symbols Used

TABLE I.

Parameters and Symbols Used

Parameters Estimated:

KLFIX	Fixed key-value length
BFI	Blocking factor (key-value index block)
RSI	Reserve space (key-value index block)
SRI	Storage requirement (key-value index blocks)
BFA	Blocking factor (accession block)
SRA	Storage requirement (accession blocks)
BFD	Blocking factor (data block)
RSD	Reserve space (data block)
SRD	Storage requirement (data blocks)
LSTAVE	Average list length for the NKEY keys
X_A	Average number of accession blocks required to hold the average list length
X_D	Average number of data blocks accessed per query
TOTSR	Total storage requirement
ACCTM	Average access time

Query Transaction Characteristics:

ACI	Average number of atomic conditions per item condition (ACI = 5)
ICR	Average number of item conditions per record condition (ICR = 4)

RCQ	Average number of record condition per query condition (RCQ = 8)
NKEY	Number of access key names inverted

Data Base Statistics:

NREC	Number of records
RLAVE	Average record length
KLAVE	Average key-value length for the NKEY keys
NVAL	Number of unique key-values, Ith key
DKV	Total number of unique key-values for the NKEY keys

TABLE II.

Device Parameters and Symbols Used

BLOCKC	Block length in characters, 10740 characters
BLOCKW	Block length in words, 1790 words
WORD	Word length in characters, 6 characters
T_T	Average time to access a block, track or page, 100 msec
T_C	Average time to compare an access key (key-value), 1.5 msec
T_I	Average time to intersect two blocks of accession numbers (pointers to records), 50 msec

TABLE III.

Parameters Used in Cost Equations

Core Size/sec.:	128 K
No. of Years:	5 Years
No. of Months:	12 Months
No. of Days:	25 Days
No. of Seconds:	3600 Sec.
Cost per Processors:	$5,000.00
Cost per Track:	0.0001¢
Access Time/sec.:	(Calculated by equations 3.15 through 3.18)
Number of Processors:	(Given by System's Architecture)
Number of Tracks:	(Calculated by equation 3.11)

APPENDIX B

Summaries of Data Base Characteristics

DESC. \ D B	1	2	3	4	5	6	7	8
NREC	3676	3676	5239	18573	15888	1296	1296	15888
NKEY	6	4	4	4	6	4	4	6
DKV	2914	496	483	1175	271	466	466	271
LSTAVE	8	30	43	63	352	11	11	352
KLAVE	10	3	4	3	4	4	4	4
RLAVE	87	87	93	84	236	404	236	404
KLFIX	2	1	1	1	1	1	1	1
BFI	447	596	596	596	596	596	596	596
RSI	45	60	60	60	60	60	60	60
SRI	8	1	1	3	1	1	1	1
BFA	1790	1790	1790	1790	1790	1790	1790	1790
RSA	179	179	179	179	179	179	179	179
SRA	14	10	14	47	60	4	4	60
BFD	123	123	115	127	45	26	45	26
RSD	13	13	12	13	5	3	5	3
SRD	34	34	51	163	398	57	33	691
TOTSR	57	46	67	214	460	63	39	753

TABLE B.1

Data Base Characteristics and Results for "Normal" Group

DESC. \ D B	1	2	3	4	5	6	7	8
NREC	800	800	800	800	800	800	800	800
NKEY	6	4	4	4	6	4	4	6
DKV	2914	496	483	1175	271	466	466	271
LSTAVE	2	6	7	3	18	7	7	18
KLAVE	10	3	4	3	4	4	4	4
RLAVE	87	87	93	84	236	404	236	404
KLFIX	2	1	1	1	1	1	1	1
BFI	447	596	596	596	596	596	596	596
RSI	45	60	60	60	60	60	60	60
SRI	8	1	1	3	1	1	1	1
BFA	1790	1790	1790	1790	1790	1790	1790	1790
RSA	179	179	179	179	179	179	179	179
SRA	3	2	2	2	3	2	2	3
BFD	123	123	115	127	45	26	45	26
RSD	13	13	12	13	5	3	5	3
SRD	8	8	8	8	20	35	20	35
TOTSR	20	12	12	14	25	39	24	40

TABLE B.2

Data Base Characteristics and Results for "Small" Group

DESC. \ D B	1	2	3	4	5	6	7	8
NREC	51000	51000	51000	51000	51000	51000	51000	51000
NKEY	6	4	4	4	6	4	4	6
DKV	2914	496	483	1175	271	466	466	271
LSTAVE	105	411	422	174	1129	438	438	1129
KLAVE	10	3	4	3	4	4	4	4
RLAVE	87	87	93	84	236	404	236	404
KLFIX	2	1	1	1	1	1	1	1
BFI	447	596	596	596	596	596	596	596
RSI	45	60	60	60	60	60	60	60
SRI	8	1	1	3	1	1	1	1
BFA	1790	1790	1790	1790	1790	1790	1790	1790
RSA	179	179	179	179	179	179	179	179
SRA	190	127	127	127	190	127	127	190
BFD	123	123	115	127	45	26	45	26
RSD	13	13	12	13	5	3	5	3
SRD	464	464	496	448	1275	2218	1275	2218
TOTSR	663	593	625	579	1467	2347	1404	2410

TABLE B.3

Data Base Characteristics and Results for "Large" Group

DESC. \ D B	1	2	3	4	5	6	7	8
NREC	3676	3676	5239	18573	15888	1296	1296	15888
NKEY	6	4	4	4	6	4	4	6
DKV	2914	496	483	1175	271	466	466	271
LSTAVE	8	30	43	63	352	11	11	352
KLAVE	10	3	4	3	4	4	4	4
RLAVE	20	20	20	20	20	20	20	20
KLFIX	2	1	1	1	1	1	1	1
BFI	447	596	596	596	596	596	596	596
RSI	45	60	60	60	60	60	60	60
SRI	8	1	1	3	1	1	1	1
BFA	1790	1790	1790	1790	1790	1790	1790	1790
RSA	179	179	179	179	179	179	179	179
SRA	14	10	14	47	60	4	4	60
BFD	537	537	537	537	537	537	537	537
RSD	54	54	54	54	54	54	54	54
SRD	8	8	11	39	33	3	3	33
TOTSR	31	20	27	90	95	9	9	95

TABLE B.4

Data Base Characteristics and Results for "Short" Group

DESC. \ D B	1	2	3	4	5	6	7	8
NREC	3676	3676	5239	18573	15888	1296	1296	15888
NKEY	6	4	4	4	6	4	4	4
DKV	2914	496	483	1175	271	466	466	271
LSTAVE	8	30	43	63	352	11	11	352
KLAVE	10	3	4	3	4	4	4	4
RLAVE	800	800	800	800	800	800	800	800
KLFIX	2	1	1	1	1	1	1	1
BFI	447	596	596	596	596	596	596	596
RSI	45	60	60	60	60	60	60	60
SRI	8	1	1	3	1	1	1	1
BFA	1790	1790	1790	1790	1790	1790	1790	1790
RSA	179	179	179	179	179	179	179	179
SRA	14	10	14	47	60	4	4	60
BFD	13	13	13	13	13	13	13	13
RSD	2	2	2	2	2	2	2	2
SRD	335	335	447	1689	1445	118	118	1445
TOTSR	358	347	493	1740	1507	124	124	1507

TABLE B.5

Data Base Characteristics and Results for "Long" Group

APPENDIX C

Access Times and Costs for Query Complexities 1, 2 and 3

ACCTM \ DB	1	2	3	4	5	6	7	8
1	1.00	2.31	3.25	5.57	23.68	1.33	1.27	27.89
2	3.32	4.37	6.04	14.99	40.34	4.58	3.72	64.70
3	6.08	7.13	8.80	17.75	43.09	7.34	6.48	67.46

TABLE C.1

Centralized Access Times for Query Complexities 1,2 and 3

"Normal" Group

ACCTM \ DB	1	2	3	4	5	6	7	8
1	0.47	0.77	0.78	0.56	1.50	0.94	0.90	1.71
2	1.55	1.80	1.80	1.69	2.99	3.22	2.67	4.25
3	4.31	4.56	4.56	4.45	5.75	5.98	5.43	7.00

TABLE C.2

Centralized Access Times for Query Complexities 1,2 and 3

"Small" Group

ACCTM \ DB	1	2	3	4	5	6	7	8
1	9.72	27.60	28.76	14.72	75.24	40.04	37.37	88.81
2	32.47	46.87	49.92	39.38	126.99	140.16	105.62	205.43
3	35.23	49.62	52.67	42.14	129.75	142.91	108.38	208.18

TABLE C.3

Centralized Access Times for Query Complexities 1,2 and 3

"Large" Group

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.82	1.09	1.39	3.46	3.61	0.61	0.61	3.61
2	1.81	1.81	2.11	4.92	4.31	1.31	1.31	4.31
3	4.57	4.57	4.87	7.68	7.07	4.07	4.07	7.07

TABLE C.4

Centralized Access Times for Query Complexities 1,2 and 3

"Short" Group

DB / ACCTM	1	2	3	4	5	6	7	8
1	1.06	3.15	4.46	6.52	31.54	1.37	1.37	31.54
2	4.60	13.01	18.46	29.86	102.75	5.46	5.46	102.75
3	7.36	15.76	21.22	32.62	105.51	8.22	8.22	105.51

TABLE C.5

Centralized Access Times for Query Complexities 1,2 and 3

"Long" Group

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.40	0.56	0.68	0.97	3.23	0.44	0.43	3.76
2	1.30	1.43	1.64	2.76	5.93	1.46	1.35	8.98
3	3.96	4.09	4.30	5.43	8.59	4.12	4.01	11.63

TABLE C.6

Concurrent Access Times for Query Complexities 1,2 and 3
"Normal" Group (8 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.40	0.56	0.68	0.97	3.23	0.44	0.43	3.76
2	0.81	0.94	1.15	2.27	5.44	0.97	0.86	8.48
3	2.01	2.13	2.33	3.46	6.62	2.15	2.04	9.67

TABLE C.7

Concurrent Access Times for Query Complexities 1,2 and 3
"Normal" Group (12 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.40	0.56	0.68	0.97	3.23	0.44	0.43	3.76
2	0.81	0.94	1.15	2.27	5.44	0.97	0.86	8.48
3	1.61	1.74	1.95	3.07	6.24	1.77	1.66	9.28

TABLE C.8

Concurrent Access Times for Query Complexities 1,2 and 3
"Normal" Group (32 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.40	0.56	0.68	0.97	3.23	0.44	0.43	3.76
2	0.81	0.94	1.15	2.27	5.44	0.97	0.86	8.48
3	1.61	1.74	1.95	3.07	6.24	1.77	1.66	9.28

TABLE C.9

Concurrent Access Times for Query Complexities 1,2 and 3
"Normal" Group (72 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.33	0.37	0.37	0.34	0.46	0.39	0.38	0.48
2	1.08	1.11	1.11	1.10	1.26	1.29	1.22	1.42
3	3.74	3.77	3.77	3.76	3.92	3.95	3.88	4.08

TABLE C.10

Concurrent Access Times for Query Complexities 1,2 and 3
"Small" Group (8 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.33	0.37	0.37	0.34	0.46	0.39	0.38	0.48
2	0.59	0.62	0.62	0.61	0.77	0.80	0.73	0.93
3	1.79	1.80	1.80	1.80	1.95	1.98	1.91	2.11

TABLE C.11

Concurrent Access Times for Query Complexities 1,2 and 3
"Small" Group (12 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.33	0.37	0.37	0.34	0.46	0.39	0.38	0.48
2	0.59	0.62	0.62	0.61	0.77	0.80	0.73	0.93
3	1.39	1.42	1.42	1.41	1.57	1.60	1.53	1.73

TABLE C.12

Concurrent Access Times for Query Complexities 1,2 and 3

"Small" Group (32 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.33	0.37	0.37	0.34	0.46	0.39	0.38	0.48
2	0.59	0.62	0.62	0.61	0.77	0.80	0.73	0.93
3	1.39	1.42	1.42	1.41	1.57	1.60	1.53	1.73

TABLE C.13

Concurrent Access Times for Query Complexities 1,2 and 3

"Small" Group (72 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	1.49	3.72	3.87	2.11	9.67	5.28	4.94	11.37
2	4.95	6.75	7.13	5.81	16.76	18.41	14.09	26.57
3	7.61	9.40	9.78	8.48	19.42	21.06	16.75	29.22

TABLE C.14

Concurrent Access Times for Query Complexities 1,2 and 3

"Large" Group (8 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	1.49	3.72	3.87	2.11	9.67	5.28	4.94	11.37
2	4.46	6.25	6.64	5.32	16.27	17.92	13.06	26.07
3	5.65	7.44	7.82	6.51	17.45	19.10	14.78	27.26

TABLE C.15

Concurrent Access Times for Query Complexities 1,2 and 3
"Large" Group (12 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	1.49	3.72	3.87	2.11	9.67	5.28	4.94	11.37
2	4.46	6.25	6.64	5.32	16.27	17.92	13.06	26.07
3	5.26	7.05	7.44	6.12	17.07	18.72	14.40	26.87

TABLE C.16

Concurrent Access Times for Query Complexities 1,2 and 3
"Large" Group (32 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	1.49	3.72	3.87	2.11	9.67	5.28	4.94	11.37
2	4.46	6.25	6.64	5.32	16.27	17.92	13.06	26.07
3	5.26	7.05	7.44	6.12	17.07	18.72	14.40	26.87

TABLE C.17

Concurrent Access Times for Query Complexities 1,2 and 3
"Large" Group (72 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.38	0.41	0.44	0.70	0.72	0.35	0.35	0.72
2	1.11	1.11	1.15	1.50	1.43	1.05	1.05	1.43
3	3.78	3.77	3.81	4.17	4.08	3.71	3.71	4.08

TABLE C.18

Concurrent Access Times for Query Complexities 1,2 and 3

"Short" Group (8 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.38	0.41	0.44	0.70	0.72	0.35	0.35	0.72
2	0.63	0.62	0.66	1.01	0.94	0.56	0.56	0.94
3	1.82	1.81	1.84	2.20	2.12	1.74	1.74	2.12

TABLE C.19

Concurrent Access Times for Query Complexities 1,2 and 3

"Short" Group (12 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.38	0.41	0.44	0.70	0.72	0.35	0.35	0.72
2	0.63	0.62	0.66	1.01	0.94	0.56	0.56	0.94
3	1.43	1.42	1.46	1.81	1.74	1.36	1.36	1.74

TABLE C.20

Concurrent Access Times for Query Complexities 1,2 and 3

"Short" Group (32 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.38	0.41	0.44	0.70	0.72	0.35	0.35	0.72
2	0.63	0.62	0.66	1.01	0.94	0.56	0.56	0.94
3	1.43	1.42	1.46	1.81	1.74	1.36	1.36	1.74

TABLE C.21
Concurrent Access Times for Query Complexities 1,2 and 3
"Short" Group (72 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.41	0.66	0.83	1.09	4.21	0.44	0.44	4.21
2	1.46	2.51	3.19	4.62	13.74	1.57	1.57	13.74
3	4.13	5.17	5.85	7.29	16.39	4.23	4.23	16.39

TABLE C.22
Concurrent Access Times for Query Complexities 1,2 and 3
"Long" Group (8 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.41	0.66	0.83	1.09	4.21	0.44	0.44	4.21
2	0.97	2.02	2.70	4.13	13.24	1.08	1.08	13.24
3	2.17	3.20	3.89	5.32	14.42	2.26	2.26	14.42

TABLE C.23
Concurrent Access Times for Query Complexities 1,2 and 3
"Long" Group (12 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.41	0.66	0.83	1.09	4.21	0.44	0.44	4.21
2	0.97	2.02	2.70	4.13	13.24	1.08	1.08	13.24
3	1.77	2.82	3.50	4.93	14.04	1.88	1.88	14.04

TABLE C.24

Concurrent Access Times for Query Complexities 1,2 and 3

"Long" Group (32 Processors)

DB / ACCTM	1	2	3	4	5	6	7	8
1	0.41	0.66	0.83	1.09	4.21	0.44	0.44	4.21
2	0.97	2.02	2.70	4.13	13.24	1.08	1.08	13.24
3	1.77	2.82	3.50	4.93	14.04	1.88	1.88	14.04

TABLE C.25

Concurrent Access Times for Query Complexities 1,2 and 3

"Long" Group (72 Processors)

COST \ DB	1	2	3	4	5	6	7	8
1	0.14	0.32	0.46	0.88	4.34	0.19	0.17	5.93
2	0.47	0.62	0.87	2.38	7.39	0.65	0.52	13.76
3	0.87	1.01	1.26	2.81	7.90	1.05	0.91	14.34

TABLE C.26
Cost for Query Complexities 1,2 and 3 "Normal" Group (Centralized)

COST \ DB	1	2	3	4	5	6	7	8
1	0.06	0.10	0.10	0.07	0.21	0.13	0.12	0.24
2	0.21	0.25	0.25	0.23	0.41	0.45	0.37	0.60
3	0.60	0.63	0.63	0.61	0.80	0.84	0.75	0.99

TABLE C.27
Cost for Query Complexities 1,2 and 3 "Small" Group (Centralized)

COST \ DB	1	2	3	4	5	6	7	8
1	1.98	5.43	5.74	2.87	21.37	14.90	10.38	33.60
2	6.61	9.21	9.97	7.69	36.07	52.14	29.34	77.72
3	7.17	9.76	10.52	8.23	36.85	53.17	30.10	78.77

TABLE C.28
Cost for Query Complexities 1,2 and 3 "Large" Group (Centralized)

COST \ DB	1	2	3	4	5	6	7	8
1	0.10	0.15	0.19	0.50	0.53	0.08	0.08	0.53
2	0.25	0.25	0.29	0.72	0.63	0.18	0.18	0.63
3	0.64	0.63	0.68	1.12	1.03	0.56	0.56	1.03

TABLE C.29
Cost for Query Complexities 1,2 and 3 "Short" Group (Centralized)

DB / COST	1	2	3	4	5	6	7	8
1	0.18	0.54	0.83	1.37	9.08	0.20	0.20	9.08
2	0.79	2.23	3.44	6.31	29.60	0.81	0.81	29.60
3	1.27	2.71	3.96	6.89	30.39	1.23	1.23	30.39

TABLE C.30
Cost for Query Complexities 1,2 and 3 "Long" Group (Centralized)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.07	0.09	0.13	0.45	0.05	0.05	0.53
2	0.17	0.19	0.22	0.37	0.82	0.19	0.18	1.27
3	0.53	0.55	0.58	0.74	1.19	0.55	0.54	1.65

TABLE C.31
Cost for Query Complexities 1,2 and 3 "Normal" Group (8 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.07	0.09	0.13	0.45	0.05	0.05	0.53
2	0.07	0.08	0.10	0.20	0.51	0.08	0.07	0.82
3	0.18	0.19	0.21	0.31	0.62	0.19	0.18	0.94

TABLE C.32
Cost for Query Complexities 1,2 and 3 "Normal" Group (12 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.07	0.09	0.13	0.45	0.05	0.05	0.53
2	0.07	0.08	0.10	0.20	0.51	0.08	0.07	0.82
3	0.05	0.05	0.06	0.11	0.23	0.06	0.05	0.38

TABLE C.33
Cost for Query Complexities 1,2 and 3 "Normal" Group (32 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.07	0.09	0.13	0.45	0.05	0.05	0.53
2	0.07	0.08	0.10	0.20	0.51	0.08	0.07	0.82
3	0.05	0.05	0.06	0.11	0.23	0.06	0.05	0.38

TABLE C.34
Cost for Query Complexities 1,2 and 3 "Normal" Group (72 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.04	0.05	0.05	0.04	0.06	0.05	0.05	0.06
2	0.14	0.15	0.15	0.14	0.17	0.17	0.16	0.19
3	0.50	0.51	0.51	0.50	0.53	0.53	0.52	0.55

TABLE C.35
Cost for Query Complexities 1,2 and 3 "Small" Group (8 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.04	0.05	0.05	0.04	0.06	0.05	0.05	0.06
2	0.05	0.05	0.05	0.05	0.06	0.07	0.06	0.08
3	0.16	0.16	0.16	0.16	0.17	0.17	0.17	0.19

TABLE C.36
Cost for Query Complexities 1,2 and 3 "Small" Group (12 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.04	0.05	0.05	0.04	0.06	0.06	0.05	0.06
2	0.05	0.05	0.05	0.05	0.06	0.07	0.06	0.08
3	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05

TABLE C.37
Cost for Query Complexities 1,2 and 3 "Small" Group (32 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.04	0.05	0.05	0.04	0.06	0.05	0.05	0.06
2	0.05	0.05	0.05	0.05	0.06	0.07	0.06	0.08
3	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05

TABLE C.38
Cost for Query Complexities 1,2 and 3 "Small" Group (72 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.21	0.52	0.54	0.29	1.44	0.83	0.73	1.80
2	0.70	0.95	1.00	0.81	2.50	2.91	2.10	4.22
3	1.07	1.32	1.38	1.19	2.90	3.33	2.49	4.65

TABLE C.39
Cost for Query Complexities 1,2 and 3 "Large" Group (8 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.21	0.52	0.54	0.29	1.44	0.83	0.73	1.80
2	0.43	0.60	0.63	0.50	1.70	2.03	1.41	2.97
3	0.54	0.71	0.75	0.62	1.82	2.16	1.53	3.11

TABLE C.40
Cost for Query Complexities 1,2 and 3 "Large" Group (12 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.21	0.52	0.54	0.29	1.44	0.83	0.73	1.80
2	0.43	0.60	0.63	0.50	1.70	2.03	1.41	2.97
3	0.21	0.28	0.29	0.24	0.82	1.07	0.68	1.55

TABLE C.41
Cost for Query Complexities 1,2 and 3 "Large" Group (32 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.21	0.52	0.54	0.29	1.44	0.83	0.73	1.80
2	0.43	0.60	0.63	0.50	1.70	2.03	1.41	2.97
3	0.21	0.28	0.29	0.24	0.82	1.07	0.68	1.55

TABLE C.42
Cost for Query Complexities 1,2 and 3 "Large" Group (72 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.05	0.06	0.09	0.09	0.04	0.04	0.09
2	0.15	0.15	0.15	0.20	0.19	0.14	0.14	0.19
3	0.51	0.51	0.51	0.56	0.55	0.50	0.50	0.55

TABLE C.43
Cost for Query Complexities 1,2 and 3 "Short" Group (8 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.05	0.06	0.09	0.09	0.04	0.04	0.09
2	0.05	0.05	0.06	0.09	0.08	0.05	0.05	0.08
3	0.16	0.16	0.16	0.20	0.19	0.15	0.15	0.19

TABLE C.44
Cost for Query Complexities 1,2 and 3 "Short" Group (12 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.05	0.06	0.09	0.09	0.04	0.04	0.09
2	0.05	0.05	0.06	0.09	0.08	0.05	0.05	0.08
3	0.04	0.04	0.04	0.06	0.06	0.04	0.04	0.06

TABLE C.45
Cost for Query Complexities 1,2 and 3 "Short" Group (32 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.05	0.06	0.09	0.09	0.04	0.04	0.09
2	0.05	0.05	0.06	0.09	0.08	0.05	0.05	0.08
3	0.04	0.04	0.04	0.06	0.06	0.04	0.04	0.06

TABLE C.46
Cost for Query Complexities 1,2 and 3 "Short" Group (72 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.09	0.11	0.15	0.63	0.06	0.06	0.63
2	0.20	0.34	0.44	0.65	2.06	0.22	0.21	2.06
3	0.57	0.71	0.81	1.03	2.46	0.58	0.57	2.46

TABLE C.47
Cost for Query Complexities 1,2 and 3 "Long" Group (8 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.09	0.11	0.15	0.63	0.06	0.06	0.63
2	0.09	0.18	0.25	0.44	1.39	0.10	0.09	1.39
3	0.20	0.29	0.36	0.57	1.51	0.21	0.20	1.51

TABLE C.48
Cost for Query Complexities 1,2 and 3 "Long" Group (12 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.09	0.11	0.15	0.63	0.06	0.06	0.63
2	0.09	0.18	0.25	0.44	1.39	0.10	0.09	1.39
3	0.06	0.10	0.13	0.25	0.68	0.06	0.06	0.68

TABLE C.49
Cost for Query Complexities 1,2 and 3 "Long" Group (32 Processors)

DB / COST	1	2	3	4	5	6	7	8
1	0.05	0.09	0.11	0.15	0.63	0.06	0.06	0.63
2	0.09	0.18	0.25	0.44	1.39	0.10	0.09	1.39
3	0.06	0.10	0.13	0.25	0.68	0.06	0.06	0.68

TABLE C.50
Cost for Query Complexities 1,2 and 3 "Long" Group (72 Processors)

ANALYSIS OF CONCURRENCY IN DATA BASE SYSTEMS

by

FERNANDO ISAAC SILVA LOPEZ

B. S., Escuela Superior de Ingenieria Mecanica y Electrica
Instituto Politecnico Nacional (Mexico), 1974

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Departament of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

Abstract

As computer users become more familiar with the capabilities of the equipment, they prefer to be responsible for their own activities - that is, operate their own equipment (under their own sets of priorities and schedules, within their own budget) and use their own data bases. Since today's computers - particularly microcomputers and minicomputers - have reached the point where the cost of acquisition is small, many business organizations which were working under centralized processing are now changing to distributed data processing.

In this report data base characteristics and system architecture are the two main factors under study. Since these two factors greatly determine cost and throughput, it is intended to present here a comparison between centralized and concurrent processing in order to see how the performance in response-time and the benefits in cost are affected when data-bases with different characteristics are used in both centralized and concurrent processing.