A SIMULATION OF A MINICOMPUTER-BASED DATA BASE TRANSACTION SYSTEM

Lawrence K. Fried
David Pravidlo

ABSTRACT

This paper is a presentation of a model which simulates an on-line minicomputer-based information system dealing with the installation and maintenance of private line circuits. The model was designed and implemented subject to several user objectives. An important objective is to identify serious bottlenecks early in system development. If such bottlenecks are identified early, the cost involved in relieving the congestion could be minimized. Moreover the model is used to study the effects of parameter and design changes on system performance and to evaluate the viability of the system.

INTRODUCTION

The system considered here was designed to mechanize operations related to the ordering, equipment allocation, preservice testing, and in-service maintenance of private line circuits. A large data base of circuit orders, circuit layouts, equipment lists and trouble ticket information is maintained. Craft personnel in Serving Test Centers (STCs) access and update this information by typing in 3-digit command codes at CRT terminals.

Among the functions which are supported are the coordination of circuit orders and service orders, assignment of central office equipment, specification of preservice tests, and the administration of trouble tickets, equipment repair and customer rebates.

During system development it was determined that more information was needed to evaluate system performance. To investigate this situation, a simulation model was developed encompassing all design aspects of the system. The simulator, which is written in GPSS, controls the progress of a transaction from the time it is entered at a terminal by a user until a response is returned to the screen. As messages are entered from regional terminals the model logic supervises their progress through the system. The delays, bottlenecks and contention are simulated through time. The speed at which messages travel through the system is referred to as system response.

MODEL STRUCTURE

The model is comprised of three major subsystems - the User and Network Configuration, the Communications Network Controller (CNC), and the Filing System. Each subsystem is modeled independently and linked through a set of defined interfaces. The subsystem approach was implemented to permit designers to:

1) alter design in one subsystem without affecting other subsystems;

2) take measurements on individual subsystems to aid in problem isolation;

and,

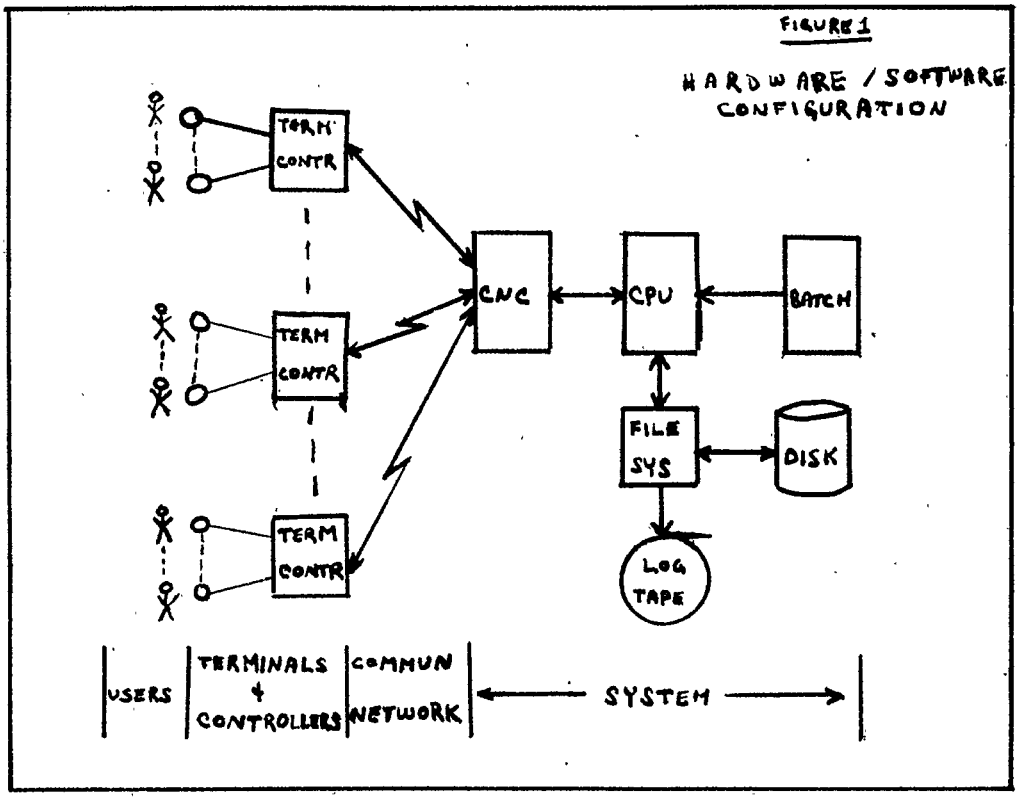
3) relate the model subsystems to the actual major subsystems for easy comparison.

The modeled Hardware/Software Configuration is illustrated in Figure 1.

USER AND NETWORK CONFIGURATION

The user model is an attempt to accurately estimate user characteristics for an entire region. An in-depth study was undertaken to define the system user in terms of transaction input rates. This involved a detailed evaluation of data provided by a representative STC. The initial task was to determine a commonality between the various work activities in today's manual environment and similar mechanized work functions in the computerized system. By analyzing the available data, system work functions were assigned frequencies of occurrence per serving link per hour. Data containing "troubles per 100 serving links" were gathered for the control office and each associated regional office. Normalizing factors were computed by forming the ratios of troubles found in the regional office to troubles found in the control office. The resulting factors were applied to the initially computed frequencies per serving link per hour for the control office to obtain similar frequencies for each regional office.

User transactions are grouped into general message types referred to as Functional Transaction Groups (FTGs). The transactions (subtasks) comprising an



FTG represent the set of commands required to implement the FTG. A total of 16 FTGs and their subtask complements are incorporated in the user model. An example of an FTG is the Order Administration Tracking function. Two subtasks associated with this FTG are:

1) Jeopardy/past due work list.

2) Activate order.

Work functions are defined as the sectionalized tasks performed in an STC. These are:

1) Dispatcher (D),

2) Equipment Administration Center (EAC),

3) Maintenance Test Center (MTC),

4) Order Administration Center (OAC):

   a) manual entry and order tracking,

   b) manual entry of associated company data,

5) Repair and Installation Center (RIC).

For each STC in the region a set of applicable work functions is defined. Since FTGs arrive at work functions with a given frequency expressed on a per serving link basis, STC serving links are apportioned by work function. The user model contains the intelligence relating FTGs and work functions. Table 1 illustrates these relationships for office 1 and office 2.

Terminal groups have been defined for each work function in each regional STC. As FTGs are assigned to work functions, they are next assigned to a free terminal associated with that STC's work function.

The initial region consists of 21 data lines and 236 CRTs located in 10 STCs. Each STC is assigned a specific number of data lines to the central computer. Each data line is associated with a terminal controller device. The terminal controller is the location where transactions generated by associated terminals wait for the line to become free.

Data representing standard subtask characteristics have been furnished and loaded into the user subsystem. Each subtask carries predefined transmit and receive character sets. The number of characters associated with the 2-way transmission of a subtask is used to determine the transmission time to and from the central computer.

The data includes the intersubtask delay time. The delay time (i.e., the average time between the receipt of a screen response and user entry of a command initiating the current subtask) has been estimated for each subtask and incorporated into the model. These delay times are necessary to simulate man/machine interfaces.

TABLE 1

Regional Configuration (For Simulation Model)

TC (Total CRT)	Time Zone	STC	WORK FUNCTIONS				
			Type	SL/WF	#CRT	Valid FTG	TG
#1 (11)	E	Office 1	O(T)	2,206	2	2,12,15,16	1
			E	2,205	1	3,12	2
			M	2,058	7	5,7,8,9,10,13	3
			D	7,593	1	6,13	4
#2 (13)	E	Office 1	O(T)	2,204	2	2,12,15,16	5
			O(ME)	3,257	1	1,12,14	6
			E	2,205	1	3,12	7
			R	4,410	1	4,11,12	8
			M	2,352	8	5,7,8,9,10,13	9
#3 (16)	C	Office 2	O	2,304	3	2,12,15,16	10
			E	2,304	1	3,12	11
			R	5,760	1	4,11,12	12
			M	2,215	10	5,7,8,9,10,13	13
			D	3,840	1	6,13	14
#4 (16)	C	Office 2	O	2,304	3	2,12,15,16	15
			E	2,304	1	3,12	16
			R	5,759	1	4,11,12	17
			M	2,215	10	5,7,8,9,10,13	18
			D	3,840	1	6,13	19
#5 (15)	C	Office 2	O	2,304	3	2,12,15,16	20
			E	2,304	1	3,12	21
			M	2,215	10	5,7,8,9,10,13	22
			D	3,839	1	6,13	23

The arrival of an FTG at a terminal is governed by a Poisson arrival rate distribution. Upon the arrival of an FTG at a terminal, each subtask comprising the FTG is sequentially entered with a predefined frequency. These frequencies were estimated by individuals with vast STC experience.

CNC SUBSYSTEM

Man/machine interaction is controlled by a multibuffered manager called CNC. This manager's ability to service many users simultaneously, overlapping I/O and computation, provides multiterminal capability, CNC's polling schemes, buffering algorithms and scheduling algorithms are modeled in detail.

As terminal transactions are entered at a CRT, they are placed in a terminal controller (TC) queue. TC queues are formed at each of the 21 terminal controllers. Transactions remain at the TC until they are removed by the polling manager.

The system has 16 core resident buffers where transaction processing can occur. At the end of a prescribed interval of clock time the buffer manager is activated. The manager scans each buffer to determine whether or not there has been a buffer status change since the last scan. A buffer must be in one of the following four states at any time:

0 - free
1 - receiving
2 - processing
3 - transmitting

The buffer manager scans the first buffer and finds it is free. It then attempts to obtain work for it. This is accomplished by activating the polling manager. This manager scans the next in line TC for queued transactions. If there are no transactions waiting at this TC, a scan is made of the next TC. This polling process continues until either of the following two conditions occurs:

1) All TCs have been scanned and no work was found.

2) A TC is scanned with at least one transaction waiting.

The first case implies that during this buffer scan phase all free buffers will remain free. In the second case, the buffer manager sets the buffer status to 1 and proceeds to send the longest waiting transaction through the data line and into the buffer for processing. The buffer manager proceeds to scan each buffer and take one of the seven actions described in Table 2. After all buffers are scanned and all required polling is complete the buffer manager becomes inactive for the prescribed time period. This complete scan is performed in effectively zero simulation time. At the end of each inactive time period the manager is activated.

TABLE 2

Buffer Scan Algorithm

Buffer Status	Buffer Completion Code	Action	End of Scan Buffer Status	End of Scan Completion Code	Description
0	0	Poll for work	0	0	Buffer was free; polling finds no work
0	0	Poll for work	1	0	Buffer was free; polling finds work
1	0	None	1	0	Buffer in RCVE status; transmission incomplete
1	1	Prepare for processing	2	0	Receive complete; prepare for processing
2	0	None	2	0	Processing in progress
2	1	Prepare for transmission	3	0	Processing complete; prepare for transmission
3	0	None	3	0	Transmission incomplete

The processing of transactions is controlled by the operating system. This operating system controls the CPU contention and the Filing System contention. Three system transaction types have been modeled. They are:

1) CNC,

2) Terminal,

3) Batch.

The system designers have assigned relative priorities to these transactions in the order shown with the CNC transaction at the highest priority.

The CNC transaction represents a means of modeling the overhead associated with the CNC task. Whenever a scan phase commences, as described earlier, any CPU processing currently underway by a lower priority transaction is preempted by the CNC. After a specified CPU hold time, control is returned to the original transaction to complete its processing.

The terminal transaction already discussed in detail is always initiated at a CRT with the output returning to the same device. Terminal transaction rates vary by STC. A group of terminal transactions comprise a general transaction called an FTG.

The batch transaction, the lowest priority transaction represents a scheduled batch process. An example of a batch process is the data base loading process required for all regional offices.

Batch and terminal transaction throughput are influenced by processing region contention. Each transaction requires a specific region for processing. At any point in time the system may contain more than one transaction requesting the same region for processing. Processing region queues are designed on a first-in first-out (FIFO) basis. The model has been designed with 17 processing regions.

FILING SYSTEM

The Filing System is controlled by OS-11 and has been considered as a subsystem by itself. The Filing System model simulates the processing time terminal transactions expend interacting with the filing system. The fallback/recovery system has been incorporated into the filing system design.

In order to process a transaction its filing system access characteristics are required. For all subtasks modeled, the following I/O data was provided as model inputs:

1) Number of Fixed Partition File Reads (FPRs),

2) Number of Fixed Partition File Writes (FPWs),

3) Number of Variable Partition File Reads (VPRs),

4) Number of Variable Partition File Writes (VPWs),

5) Number of Fixed Partition Read Before Writes (FRWs),

6) Number of Variable Parition Read Before Writes (VRWs),

7) Assigns,

8) Releases,

9) Allocates,

10) Deallocates.

The foregoing ten data items represent "logical" I/Os and are provided for each subtask. However, each logical I/O requires a number of physical disk accesses to achieve the "logical" result. Moreover, the processing time associated with the physical disk access is one of the key parameters of the filing system model. System designers have defined the physical I/Os associated with each logical I/O type. File System processing is modeled to satisfy one physical I/O per command at a time. However, it is possible to have more than one transaction accessing the filing system simultaneously. This concept is referred to as a reentrant filing system. The reentrancy level denoting the number of concurrently processing transactions in the filing system has been modeled as a system parameter.

Upon entering the filing system a command or transaction must first find one available copy of the filing system and an available processing region.

The processing of physical I/Os involves seeks, data transfer, and CPU processing. Queuing occurs at each disk controller. Each disk controller is associated with eight disk drives. I/Os must first seek their assigned disk drive and then perform the data transfer. Only one character transfer can be supported at a time by a disk controller while seeks can be occurring on all drives except the drive on which the character transfer is occurring. Therefore, a single controller will have a queue of users waiting to access the controller to do a character transfer. While waiting they may be having their seeks processed if the required drive is not busy.

All physical I/Os are subject to delays due to rotational latency. This is the time required for the correct data to rotate to the read/write head so the data transfer can begin.

The CPU time associated with a terminal transaction is a function of the I/O processing performed by the application software. As each physical I/O is processed a proportionate slice of CPU time is requested subject to the contention from other requests.

The concept of file system locks and unlocks represents a protective feature for the filing system. The potential for locking exists whenever a command attempts to satisfy one of the following logical I/O requests:

- Variable Partition Read,
- Variable Partition Write,
- Variable Partition Read Before Write,
- Allocate,
- Deallocate.

If another command has begun processing an allocate or a deallocate, the command attempting to process one of the above requests will be suspended and held in a queue. The held transaction(s) is then permitted to resume processing. There can be no more than one transaction concurrently locking the filing system. However, many transactions may be suspended simultaneously pending completion of the locker's remaining tasks.

Log tape processing is a system process modeled to measure the overhead of logging. The logging process is executed whenever a logical write I/O is requested and when a command has completed processing all I/Os. The records which require logging are first written to a log buffer. The logging requires the CPU for processing. When the buffer is full and no more can be written, the buffer must be dumped to tape. The dumping process requires additional CPU processing. The in-core buffer has been modeled as a double buffer. When these buffers both become full they are dumped in a 2-step process.

SIMULATION MODEL CHARACTERISTICS

The simulation model was written under GPSS version 5. The model consists of more than 300 "GPSS blocks." These blocks control the progress of user transactions as they flow through the system. The vehicle for maintaining data inherent to a transaction is the transaction parameter. The following are some of the information stored in transaction parameters:

1) originating terminal,

2) data line or TC number,

3) terminal group number,

4) FTG,

5) subtask number,

6) buffer assignment,

7) terminal to computer transmission time,

8) computer to terminal transmission time,

9) CPU service time,

10) number of FPFs, FPWs, VPRs, VPWs, etc.,

11) drive number,

12) disk controller number,

13) rotational latency time,

14) clock time subtask entered system.

Transactions are initially generated representing FTGs (see page 1). Those transactions whose arrival rates are time dependent are regenerated every hour. Those that are not time dependent are generated only once - at system startup.

Each FTG transaction generates one transaction for each terminal group handling that FTG. Each resulting transaction is assigned an arrival time at

its respective terminal group. The arrival time is based on the following:

1) a Poisson arrival distribution,

2) a mean arrival rate.

After each transaction enters the system at its computed arrival time the next appearance of this FTG type transaction at this terminal group is computed. This arrival time is computed based on the current hourly rate. However, at the start of each modeled hour an FTG transaction is regenerated for each FTG. The newly created FTG transaction assumes control of transaction generation with the new hourly rate.

At the instant an FTG is scheduled to arrive a free terminal is selected from the terminal group. This terminal becomes reserved until all subcommands associated with the FTG are satisfied. If all terminals in the terminal group are reserved (in-use) the transaction waits until one becomes free.

After a terminal is selected the first subtask is entered. When a response is received, a predefined user delay is imposed before the next subtask is entered. Before each subtask is entered the frequency input data parameter is examined. If the frequency is an integer then the subtask is sequentially entered a number of times corresponding to the frequency. If the frequency is noninteger the integer portion is extracted. The fractional portion is compared to a draw from a uniform distribution. If the draw is less than or equal to the fractional portion then one is added to the integer portion to determine the number of subtask entries. If the draw is greater than the fraction, the integer portion determines the entry count. When all subtasks have been processed the FTG processing is considered complete. The terminal is freed for use by other FTGs arriving at this terminal group.

When subtasks are entered they are placed on their respective terminal controller queues. These subtasks are subsequently removed in the order they arrived. A total of 16 buffers are available to accommodate these subtasks prior to processing. The buffer manager becomes activated once every 1000 msec. Polling occurs when the buffer manager scans these buffers and finds a free buffer. For each free buffer encountered a poll is taken. Polling is performed by beginning at the TC immediately following the last poll. When the polling list has been exhausted polling resumes at the top of the list. If all members in the list are empty (i.e., no work) polling is terminated for the remainder of this scan phase (i.e., the scan phase takes place in effectively zero time and therefore no additional work can be generated during this phase).

Each subtask released from a TC is transmitted with a predefined transmission time and subsequently resides in the assigned buffer. This is referred to as the buffer receive status. This subtask cannot commence CPU/Filing System processing until the buffer manager is reactivated (every 1000 msec) and proceeds to scan. When reactivation occurs, the manager detects the status change and resets its status to processing. The subtask is then directed to the processing subsystem (i.e., Filing System/CPU).

A subtask entering the processing subsystem must first find its required core region free. If the region is available, the subtask reserves it for its use. The current filing system reentrancy level (number of subtasks concurrently processing) is compared to the reentrancy limit. If the current level is below the limit the subtask is permitted entry. If the filing system is currently operating at the reentrancy limit the subtask is placed in a queue designated the REENT queue. Subtasks remain queued until the reentrancy level falls below the limit.

The filing system I/O requirements associated with the subtask are converted into physical I/Os. Table 3 presents the number of physical I/Os and seek type (i.e., fixed or variable) probabilities associated with each logical I/O type.

Only one physical I/O for the subtask can be processed at a time. When a physical I/O completes processing the next physical I/O is activated.

After a controller is selected, the physical I/O must perform a seek on a random drive. This drive is unavailable to other physical I/Os until the physical I/O completes its data transfer. Seeks are performed either while waiting to reach the controller head cell, at the head cell or a fraction at each. Seek times vary with file type (fixed or variable). The average number of cylinders traversed for a seek on a fixed file differs from that on a variable partition file.

Data transfer time is a function of the number of words in the record and the data transfer rate. The additional overhead of the effects of rotational latency is built into the model.

After each physical I/O completes its data transfer, it proceeds to seize the CPU and elapse some processing time. The physical I/O desiring CPU servicing is subjected to contention for the CPU.

When all physical I/Os have accomplished the required file I/Os the region is released and the buffer status is reset to processing complete. When the next scan phase occurs, the buffer scan finds this buffer ready for return transmission and therefore sets the status to transmit. When the proper data line is released the subtask is transmitted back to the user.

VALIDATION OF THE MODEL

Validation of the simulator was a difficult task due to the fact that the system was still under development at the time. The model was run as a single-threaded system, and the results were compared to measurements obtained from the actual single-user system. Response times predicted by the model were within 10 percent of actual system measurements.

TABLE 3

Required Physical I/Os and Seek Probabilities

Logical Record I/O	Physical I/Os (Worst Case)	Probability of a Fixed Seek
FPR	2	1.00
FPW	4	1.00
VPR	5	0.40
VPW	11	0.454
FRW	4	1.00
VRW	8	0.625
Assign	3	1.00
Release	3	1.00
Allocate	20	0.85
Deallocate	15	1.00

SIMULATION RUN-TIME PHILOSOPHY

The original model was run for nine 1-hour simulation periods each representing the average business day. User input rates were provided reflecting busy and nonbusy periods.

It was felt that performance characteristics representing the 9-hour day could be obtained by applying the transaction rates for the busiest hours of the day and simulating for shorter periods of time using intermediate GPSS resets. The resets were designed to minimize the transient effects.

The baseline model was found to reach a steady state after 3 hours of simulated time. System response time results compared to within 2 percent of the original 9-hour run. This resulted in substantial savings of computer run-time.

MODEL OUTPUT

A report generator was created to output those results which were considered to be most indicative of system performance. The following items of information are contained in this set of output:

I. Input Load Characteristics

- A. Distribution of transaction input rates.
- B. Distribution of transaction types (FTG 1, FTG 2, etc.).
- C. Total number of transactions processed.
- D. Total number of batch transactions.

II. System Response Characteristics

- A. Distribution of user response times.
- B. Average user response time on a per-command basis.
- C. Distribution of system response times.

III. Machine Usage Characteristics

- A. I/O profiles.
  1. Distribution of controller selection
  2. Distribution of drive selection
  3. Distribution of log tape execution times
  4. Average controller utilization
  5. Average drive utilization
  6. Queuing statistics for controllers
  7. Queuing statistics for drives
- B. CPU
  1. Average CPU utilization
  2. Queuing statistics for CPU
- C. Transmission Lines
  1. Distribution of number of characters received by CNC buffers from terminals
  2. Distribution of number of characters transmitted by CNC buffers to terminals

MODEL EXPERIMENTS

A total of 13 experiments were run to determine which parameters most affected system performance. All experiments can be compared to the original (baseline) model which had the following characteristics:

- 236 users

- unaligned files
- 4800 baud transmission rate
- filing system reentrancy level = 4
- two 4K log-tape buffers
- CNC buffer scan every second
- 17 processing regions (one dedicated to batch)
- 3 disk controllers
- 8 disk drives per controller

What follows is a list of the 13 experiments that were run:

1) Baseline Model
2) Exp. 1 with Filing System Reentrancy Level = 1
3) Exp. 1 with Filing System Reentrancy Level = 8
4) Exp. 1 with 9600 Baud Data Transmission Rate
5) Exp. 1 with Sector-Aligned Files
6) Exp. 1 with two 8K Log-Tape Buffers
7) Exp. 1 with two 16K Log-Tape Buffers
8) Exp. 1 with 128 Users
9) Exp. 1 with 30 Users
10) Exp. 1 with Single User
11) Exp.'s 3, 4, 5, 6 together
12) Exp. 11 with 2*User Input Rate
13) Exp. 11 with 4*User Input Rate

For comparison purposes, the aforementioned experiments have been grouped into the seven study categories listed below:

Case Study		Experiments
1	Filing System Reentrancy	1,2,3
2	Data Transmission Rate	1,4
3	Sector-Aligned Files	1,5
4	Size of Log-Tape Buffer	1,6,7
5	User Load	1,8,9,10
6	Optimizing System Performance	1,11
7	Increased Capacity Due to Optimization	1,11,12,13

The two measurements which are considered to be most important for evaluating system performance are system and user response times. System response time is defined as the time elapsed between the following two events:

1) the last character transmitted to the computer, and
2) the first character transmitted back to the user.

Similarly, user response is defined as the time elapsed between the following two events:

1) the enter button depressed on the CRT, and
2) the last character transmitted back to the user.

The difference between system and user response is the wait time at the terminal controller queue, and the transmission time to and from the central computer.

CASE STUDY 1: FILING SYSTEM REENTRANCY

This experiment clearly demonstrates the need for a reentrant filing system. User response times increased to over 5 minutes when the model was run with a nonreentrant filing system (Exp. 2). Nearly all of the system response time was spent waiting for a copy of the filing system. Note, however, that once a transaction gained access to the filing system, it required less processing time than did transactions in a reentrant filing system. This is due to the fact that there was no disk contention in the nonreentrant filing system. Increasing the reentrancy level to 8 resulted in a 50 percent reduction in user response time, thus making this a very desirable feature to implement (Table 4).

TABLE 4

Impact of Filing System Reentrancy

Exp.		Sys	User	Avg. Time Spent Waiting for Access to File System
1	Baseline	7.2	18.4	4.6
2	Reent Level=1	23.5	304.8	22.3
3	Reent Level=8	5.1	8.2	2.0

Sys = Avg. System Response Time (in seconds)

User = Avg. User Response Time (in seconds)

Baseline Reentrancy Level = 4

CASE STUDY 2: DATA TRANSMISSION RATE

The effect of doubling the transmission rate is examined in this experiment. As one would expect, system response time was not affected by a change in transmission rate. User response time did improve by nearly 50 percent. See Table 5.

TABLE 5

Impact of Data Transmission Rate

Exp.		Sys	User	User (90%)
1	Baseline	7.2	18.4	40
4	9600 Baud	7.2	10.4	22

Sys = same as in Table 4

User = same as in Table 4

User (90%) = T, such that 90% of user response times < T

Baseline Transmission Rate = 4800 baud

CASE STUDY 3: SECTOR-ALIGNED FILES

Alignment of files on sector boundaries allows transactions to access disk files with fewer I/Os. Thus, in the model, file alignment was simulated by reducing the number of physical I/Os necessary to implement a logical I/O (read, write or read before write for both FPF and VPF files). Sector alignment upgraded system performance more than any other single factor. Reductions in number of I/Os resulted in shorter wait times for disk controllers and disk drives. The results of this experiment are shown in Table 6.

TABLE 6

Impact of Aligning Files on Sector Boundaries

Exp.		Sys	User	Con Q	Drv Q
1	Baseline	7.2	18.4	3.9	1.7
5	File Alignment	2.6	5.5	1.2	0.7

Sys = same as in Table 4

User = same as in Table 4

Con Q = Avg. queue time for disk controller (in milliseconds)

Drv Q = Avg. queue time for disk drive (in milliseconds)

CASE STUDY 4: SIZE OF LOG-TAPE BUFFER

Increasing the size of the log-tape buffers influences response time in two ways:

1) fewer buffer dumps to the log-tape, and

2) longer CPU time for dumping a buffer to tape.

In this study, the size of the log-tape buffer was increased twice to determine the effects of these two factors on system behavior. The results (Table 7) indicate that an increase in buffer size did not have a significant impact on system performance. A buffer size of 8K yielded slightly better results than buffer sizes of 4K (baseline) and 16K. The effect and need of this factor is dependent on the write activity of the transactions.

TABLE 7

Impact of Size of Log-Tape Buffer

Exp.		Sys	User	User (90%)
1	Baseline	7.2	18.4	42
6	Two 8K Buffers	7.1	15.1	34
7	Two 16K Buffers	7.0	18.1	42

Sys = same as in Table 4

User = same as in Table 4

User (90%) = same as in Table 5

Baseline Model has two 4K Buffers

CASE STUDY 5: USER LOAD

The baseline model was run with a fully loaded region of 236 users. The purpose of this experiment is to determine the sensitivity of the system to reductions in the number of users. The initial reduction of users from 236 to 128 resulted in the greatest improvement in response time. User response time decreased by almost 2/3, suggesting that queuing for the terminal controllers does not begin to degrade response time until there are at least 128 users on the system. The results of this experiment are shown in Table 8.

TABLE 8

Impact of Reduced User Loading

Exp.		Sys	User	User (90%)
1	Baseline	7.2	18.4	42
8	128 Users	3.9	6.7	10
9	30 Users	3.0	5.9	10
10	1 User	2.0	4.7	8

Sys = same as in Table 4

User = same as in Table 4

User (90%) = same as in Table 5

Baseline Model has 236 Users

CASE STUDY 6: OPTIMIZING SYSTEM PERFORMANCE

In previous experiments, a single parameter was modified to determine what effect it had on system behavior. In this experiment all parametric changes which resulted in an upgrading of performance were incorporated into the model. The results are shown in Table 9.

TABLE 9

Impact of Multiple Parametric Modifications

Exp.		Sys	User	User (90%)
1	Baseline	7.2	18.4	42
11	Reent Level=8, 9600 Baud, File Alignment, Two 8K Log-Tape Buffers	2.4	3.9	6

Sys = same as in Table 4

User = same as in Table 4

User (90%) = same as in Table 5

CASE STUDY 7: INCREASED SYSTEM CAPACITY DUE TO OPTIMIZATION

The previous experiment showed that substantial reductions in response time could be realized through the modification of several parameters. This experiment examines the effects of an increased transaction rate on the upgraded system obtained in Exp. 11. Response times increased only slightly when the transaction rate was doubled (see Table 10), indicating that the modified system had a great deal of potential for growth. Doubling the transaction rate again did result in some degradation of user response time. In spite of this fact, the user response of the improved system at 4 times the transaction rate was still twice as fast as that of the baseline model, even though the CPU was working almost twice as hard (72 percent versus 38 percent utilization).

TABLE 10

Impact of Increasing the Transaction Rate

Exp.		Sys	User	CPU
1	Baseline	7.2	18.4	.38
11	Upgraded Sys.	2.4	3.9	.40
12	Upgraded Sys. (2*XAC Rate)	2.6	4.4	.57
13	Upgraded Sys. (4*XAC Rate)	3.0	9.4	.72

Sys = same as in Table 4

User = same as in Table 4

CPU = CPU utilization

CONCLUSIONS

Several methods of improving system performance have been investigated. The results indicate that the greatest improvements in performance can be realized by making modifications to the original design of the filing system.

There is considerable debate concerning how long a period is "acceptable" for a terminal user to wait prior to receiving a response from the system. The objective set by designers of the system was an average user response time of 5 seconds. This study shows that there exists a set of feasible modifications to the system which results in meeting this goal, and at the same time increases the capacity of the system.

OTHER BENEFITS OF THE SIMULATION EFFORT

In addition to experiments specifically aimed at reducing the response time of the system, the model may be used to evaluate proposed system redesigns. Examples of this are a reconfiguration of the disks and a change in the polling scheme used to query terminal controllers.

Because of the fact that the subsystems were modeled independently, the model can be converted to models which simulate other similar systems. An effort is currently underway to convert the model to one which simulates another system with similar versions of the CNC and filing system.

Perhaps one of the most important benefits derived from this effort was an increased understanding of system operation at the design level.

ACKNOWLEDGMENTS

The authors wish to thank Wayne Hatter of the Office Maintenance Systems Department of Bell Laboratories and Martin Goldberg of Southern Simulation Inc. for their contributions to the modeling effort.