SIMULATION OF A MULTIPLE CPU MILITARY COMMUNICATIONS SYSTEM

James G. Sprung
Ved Aggarwal

ABSTRACT

The Pentagon Telecommunications Center (PTC) provides telecommunications support to Headquarters, Department of the Army and other government agencies. The current system provides the Army with a full range of communications functions, i.e., receipt, transmission, message validation, and online CRT operations, centered around an IBM System/360 (Model 65) processor.

The Automation Support Detachment, under the U.S. Army Communications Electronics Engineering Installation Agency - CONUS, performs telecommunications systems design, development, and support for the system. The design efforts entail extending similar operational services to all communication centers in the Pentagon. The proposed message workload is expected to increase to 6000 messages per hour from the current 1500 messages per hour. During the design effort, a simulation model is being used to estimate the performance impact of proposed system configurations and design changes.

This paper describes the modeled system and some simulation experiments made using the model as a design tool. The main objectives for the experiments were:

(1) To determine the processor speed required to process 6000 messages per hour (the expected future workload).

(2) To determine whether off-loading some of the processes associated with a message's input/output (I/O) would provide a significant improvement in central processor utilization and message throughput.

(3) To investigate whether I/O delays caused by the disk subsystem would be reduced if reads were alternated between the primary and backup disk drives.

BACKGROUND

The Pentagon Telecommunications Center (PTC) provides telecommunications support for the Department of the Army Headquarters (HQDA). The PTC is based on a computer system that transmits, receives, and relays messages for HQDA and other government agencies. The Center also provides transmission and receipt of messages to and from the Automated Digital Network (AUTODIN), State Department, and various Department of Defense and Executive Branch agencies. The Center currently processes up to 1500 messages per hour with an average size of 2000 characters per message.

The PTC has been directed to acquire a joint Army, Navy, Air Force, and Joint Chiefs of Staff (JCS) message processing system. The computer system at the PTC will be expected to process 6000 messages per hour and to link the Staff Service Centers in order to service all the services.

The U.S. Army Communications Electronics Engineering Installation Agency (Automation Support Detachment [ASD]) is responsible for the design and software development of the PTC. ASD requested that FEDSIM build a model of the current and proposed systems to evaluate potential design changes associated with the additional message traffic and changes in message processing.

SYSTEM DESCRIPTION

Hardware

The PTC is currently centered around two IBM System/360 (Model 65) processors. One of the processors is used to perform the message processing functions and the other is used as a backup. A COMTEN 3670 communications interface unit is attached to the system via a byte multiplexor channel. The communications interface unit connects the system to 145 communications lines (channel designators [CD]) whose speeds vary from 75 to 9600 bits per second. The system depends heavily on two banks of Telex 6330 disk drives which are connected to the system via two selector channels.

Software

The application software consists of many core resident, reentrant modules which fall into one of eight categories:

1. Control Programs - Modules that initialize and monitor the system, and pass control to the subsystems as required.

2. Input/Output - Modules that control peripheral devices, collect data for the audit trail, and develop the message header/trailer.

3. Video Display - Modules that service requests from the video display consoles.

4. Message Processing - Modules that detect message errors and the determination of addresses.

5. Table Subsystem - Modules that build and maintain control tables.

6. Data Management Subsystem - Modules that perform data base functions of space allocation, file management, and message accounting and protection.

7. Offline Programs - Modules that respond to offline retrieval requests and that produce traffic statistics.

8. COM Subsystem - Modules that cause recording of narrative messages for placement on microfilm.

The processing of a message entails the use of modules from each of these application subsystems. Illustration 1 shows the flow of a message through the stages of message processing.

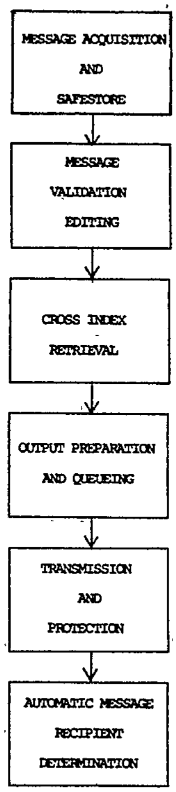
SIMULATION MODEL

FEDSIM developed a PCTCS model in the Extendable Computer System Simulator Language (ECSS II). ECSS II is a flexible computer modeling language that is capable of modeling different hardware devices, software applications, operating systems, and device interfaces.

The PCTCS model was based on the previous and current system. The model was initially baselined to the SLACMON and file use data acquired (from MEASURE software monitor data) when the system was centered around an IBM System/360 (Model 50). The model was changed to reflect such system enhancements as additional terminals, new processors, new disk drives, and new software.

ILLUSTRATION 1

Message Flow



For each change, the model was revalidated to insure consistency with system performance data collected by software monitoring. The following parameters were used to validate the baseline model:

- CPU utilization
- Processing requirements for major software functions
- Number of input message blocks processed
- Number of output message blocks processed
- Total number of I/O completions
- Total number of messages completed
- Message traffic by each Channel Designator

SIMULATION EXPERIMENTS

Experimentation with the simulation model concentrated on current and planned configurations to support the PACS. Planning called for system traffic growth from 1600 to 2400 and finally to 6000 messages per hour, additional communication lines for the transmission or reception of the messages, and incorporation of software enhancements to support additional message formats and message processing functions.

Simulation Experiment To Study Changes To The PACS Hardware Characteristics

The simulation model served as a design tool for answering questions about postulated future systems. The main processor being a critical resource, its capability and effectiveness was evaluated to process additional workload which was expected to increase with the implementation of various phases to the PACS system.

The objective of the experiment was to determine the speed of an IBM System/360 (Model 65) replacement that could handle a maximum future workload of 6000 messages per hour for the PACS system.

Table 1 shows the message and character load generated by five pseudo random generators at a 6000 message per hour traffic rate. A previous FEDSIM study indicated that variations in the message traffic volume (in terms of characters processed) influence the PACS performance significantly. Therefore, the highest character load (Set 1 - Table 1) was used for the simulation experiments which placed the greatest processing demand on the PACS system, thus giving results that tended to be "worst case" from the system performance standpoint.

TABLE 1

Comparison of Five Message Loads at 6,000 Messages Per Hour

	INPUT		OUTPUT		TOTAL	
Message Set No.	Messages	Characters	Messages	Characters	Messages	Characters
1	396	1,541,336	820	2,605,828	1,216	4,147,164
2	384	1,273,551	784	1,844,806	1,168	3,418,357
3	409	1,423,161	895	2,190,530	1,304	3,613,691
4	375	1,614,073	777	2,451,180	1,152	4,065,253
5	392	1,423,068	834	2,281,217	1,226	3,704,285

NOTE: Message loads above were created by the Message Generator Program using:

a. 6,000 messages per hour rate
b. different random number streams

The model was used to determine the PACS performance based on a 6000 message per hour traffic volume (the expected maximum workload). Three simulation runs were made with different processor speeds of 2, 2.5, and 3 times the speed of an IBM System/360 (Model 65) processor. Only one statement change was required in the System Description section of the simulation model to modify the processor speed. The performance measures used to evaluate any performance improvement were:

- CPU utilization
- Average message turnaround time
- Maximum message turnaround time
- Number of messages completed
- File utilization and queue length (particularly for the Checkpoint (DBF914) and Intransit (DBF932) files).

Three simulation runs were made (each with a different CPU speed) using the message and character load generated by Message Generator Set 1. Table 2 shows CPU utilization and message turnaround time (from receipt of first character in the PACS system until the preparation of the first message block for transmission) for each simulation run. The variations in CPU utilization as a result of variations in processor speed are illustrated in Illustration 2.

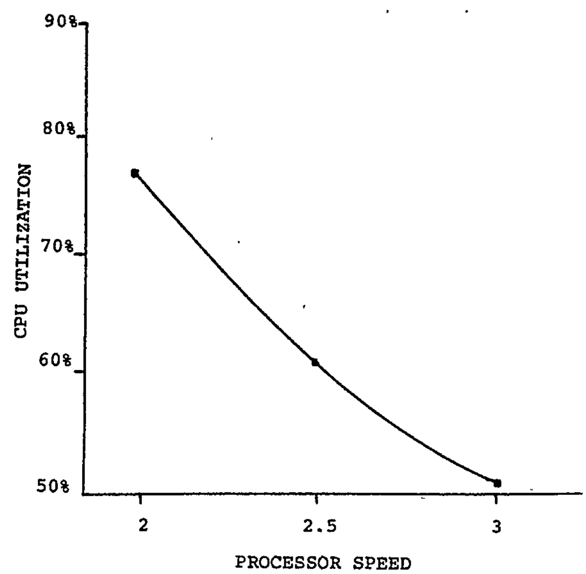
TABLE 2

System Performance For Variations in the Processor Speed

PERFORMANCE CHARACTERISTICS	RUN 1	RUN 2	RUN 3
CPU Utilization	76.5	61.1	51.0
Message Turnaround Time (seconds)			
Average	2.6	2.1	1.9
Maximum	28.1	24.8	21.1
Messages Completed			
Input	323.0	323.0	323.0
Output	672.0	672.0	673.0
Average Queue Time per File (Intransit) Access (seconds)	0.373	0.412	0.425

ILLUSTRATION 2

CPU Utilization vs Processor Speed



The results indicated that the system could handle the simulated traffic volume rate of 6000 messages per hour with no major queue buildup or effect on the message turnaround time.

Queueing statistics for two PACS files - Checkpoint and Intransit - were analyzed in detail since these files account for approximately 70% of total file accesses. No significant queue buildup nor file access delay were observed for the Checkpoint file. The average delay time per file access for the Intransit file was significant, but no I/O bottleneck nor queue buildup was observed.

The simulation results indicated that the PACS System with any of the three CPU speeds could handle the traffic volume rate of 6000 messages per hour with no major queue buildup or significant difference in the message turnaround time.

The customer used the minimum recommended speed as the basis to issue a Request For Proposal (RFP) for the procurement of the main processor. A speed of 2.2 times the speed of an IBM System/360 (Model 65) processor was used to size the processor since additional software functions were expected to be added.

Simulation Experiment To Study Enhancement To Software Functions

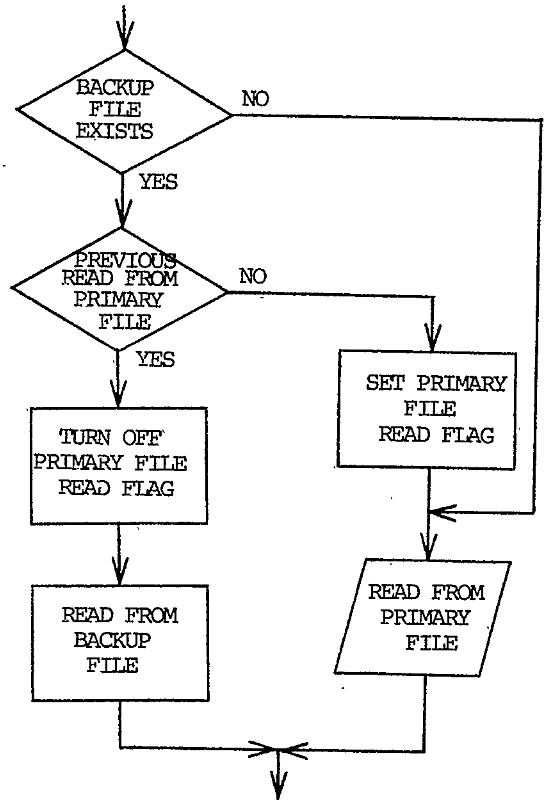
The future online data base will reside on 32 - Telex 6330 disk packs, and will be accessed through two block multiplexor channels. The records are currently written simultaneously on both the primary and backup files for reliability, but are read only from the primary files.

The objectives of this experiment were to explore whether it is possible (1) to reduce the amount of file contention and (2) to achieve balance between both channels, if the records are read alternately from the primary and backup files.

ASD provided the I/O information from the measure data (IBM 2314 disk counts) on the disk access basis only, but the disc access inputs for the model were required on the file access basis. Consequently, when the data base was changed from IBM 2314 disk packs to Telex 6330 disk packs, an imbalance in I/O disk accesses occured between the channels and among the disk drives. To resolve these inconsistencies, disk accesses were modified in the model and the file allocation in the system was changed to level the load across the Telex 6330 packs. This modifiication was validated by comparing the number of disk accesses from the model run and the Measure data. ASD provided the (July 26, 1977 to August 15, 1977) traffic data for the PACS in terms of message size and message volume for each CD. The data were utilized to develop the simulated workload of input and output messages for the PACS model. Illustration 3 is a flow chart of the logic required to implement the "Alternate Read" modification in the model.

ILLUSTRATION 3

"Alternate Read" Modification Flow



The performance measures used to evaluate any performance improvements due to the proposed "Alternate Read" modification were:

- File utilization and queue buildup (particularly for the Checkpoint DBF914 and Intransit DBF932 files)
- Channel balancing
- CPU utilization
- Average message turnaround time

Two experiments described below were performed to determine if any improvement in the utilization of system resources and in I/O operations could be expected from the implementation of the "Alternate Read" technique.

- Baseline Experiment - The system load was 2400 input messages per hour. The records were read from the primary files only.

- Alternate Read Experiment - The system load was the same as in the baseline experiment but the records were read alternately from the primary and backup files.

Analysis of simulation results indicated no significant improvement in the utilization of system resources nor in the I/O operations as described below.

1. I/O Activity - Table 3 gives the I/O activity for most active files, which account for approximately 80% of total file accesses. The simulation results shows that with an "Alternate Read" modification, a balance of I/O activity can be achieved for both the primary and backup files of Intransit and Message Control files. Since Checkpoint and Distribution are write only files, there was no change in the number of file accesses for the primary and backup files. Table 4 shows more balance in I/O interactions between the two channels. But even if a proper balance of I/O activity on files and channels is achieved, there is no significant improvement in the average access time (see Table 3). This is because at the current simulated workload, no major queues developed for the files and the channels.

TABLE 3

File I/O Activity Statistics

*FILE NAME	NUMBER OF I/O ACCESSES		AVERAGE I/O ACCESS TIME (MILLISECONDS)	
	BASE LINE	ALTERNATE READ	BASE LINE	ALTERNATE READ
DBF914A-Checkpoint	976	975	9.9950	10.0639
DBF914B	976	975	9.9848	10.0639
DBF932A-Intransit	1095	758	29.9003	28.3202
DBF932B	433	769	25.0330	28.5185
DBF907A-Distribution	138	138	39.3941	39.3213
DBF907B	138	138	39.2940	39.3210
DBF903A-Message Control	364	273	39.2276	38.9752
DBF903B	182	273	37.0036	38.8969

*File name ending with 'A' represents the primary file.
File name ending with 'B' represents the backup file.

TABLE 4

Channel Activity Statistics

CHARACTERISTICS	BASE LINE CHANNEL #1	BASE LINE CHANNEL #2	ALTERNATE READ CHANNEL #1	ALTERNATE READ CHANNEL #2
TOTAL I/O REQUESTS	3152	1868	2568	2444
REQUESTS QUEUED	545	213	381	389
AVERAGE QUEUE LENGTH	0.099	.020	0.060	0.062
MAXIMUM QUEUE LENGTH	5	5	6	6
AVERAGE QUEUE TIME (MILLISECONDS)	2.825	0.953	2.088	2.269
MAXIMUM QUEUE TIME (MILLISECONDS)	90.938	49.174	70.130	85.111

2. CPU Utilization - There was no change in CPU utilization in the alternate read experiment. The average utilization decreased from 38.72% to 38.66%, which could be attributed to the model's random behavior.

3. Message Turnaround Time - No significant change in the message turnaround time was achieved because there was no improvement in the I/O operations as discussed above. The other factors which contributed to a message completion time, e.g., sequence of operations and system resource consumption, were identical in a message for both experiments.

The preliminary simulation results indicated no significant improvement in the system performance due to the proposed "Alternate Read" software modifications, at a current simulated workload of 2400 messages per hour. This is because the baseline model experiment showed no significant queue buildup for either channel. At a higher message rate (e.g., 6000 messages per hour), significant queue buildup is expected to develop, therefore, the "Alternate Read" software modification may yield significant improvement in the system performance. However, further simulation experiments are required to confirm this impression.

Simulation Experiment To Study Off-Loading Of Software Functions

Since the PACS System must accommodate a greatly expanded message workload, higher capacity and faster equipment is being procurred. To process higher message workload, ASD is exploring ways to increase PACS message processing capability by decreasing demands for the critical system resources.

The objective of this study was to determine if significant reduction in demand for the main processor could be achieved to justify procurement of a Terminal Control Subsystem (TCS) if certain software functions were off-loaded from the main processor to a TCS or the COMTEN 3670.

To provide a basis to measure the effect of off-loading, it was necessary to run a baseline experiment without off-loading any software functions. The baseline experiment used a workload of 2470 messages per hour, a workload considered representative for the study. The hardware configuration included a CPU and TCS with the speed of an IBM System/360 (Model 65), a COMTEN 3670, disks equivalent to IBM 3330 disks, and Channel Designators whose transmission rates had been selectively increased to prevent I/O backups due to the high message rate. The software functions which were off-loaded from the CPU to the TCS or COMTEN 3670 for this study are described as follows:

1. Input Translation and Blocking Module (IOI060) - This module accepts input data in a buffer, translates the data to EBCDIC based on the input character codes, and moves the translated data to message blocks. Approximately 30% of its processing functions were off-loaded to the COMTEN 3670.

2. Main Control Module (I00530) - This module controls the flow of processes in the Data Management Interface conglomerate. It also builds line blocks for all devices (except the printer) and translates all line blocks to the appropriate line code. Approximately 40% of its processing functions were off-loaded to the COMTEN 3670.

3. Printer Line Block Build Routine (I00550) - This module initializes page frames from annotations and other sources and builds printer transmission line blocks from message data and the page frames. All of its processing functions were off-loaded to the TCS.

Since none of the off-loaded modules had any supervisor calls (SVCs) or file accesses, CPU utilization was the only statistic which showed the difference between experimental and baseline run. Since the model is driven primarily by the input message workload, some delay is experienced at the beginning of each run before output messages are generated in significant number for the model to stabilize as shown in Table 5. The model reached a stable condition during the third reporting period. The decrease in CPU utilization for the last eight intervals was 4.5 percent (from 51.47 to 46.97) which represents approximately 9 percent reduction in demand for the CPU. Table 5 also shows additional resource requirements for the TCS and COMTEN 3670. Message turnaround times for the two runs showed no significant differences, since the off-loaded modules were executed in the same sequence in both runs.

TABLE 5

Processor Utilization

	PERCENT UTILIZATION			
	BASELINE	EXPERIMENT		
MINUTE	CPU	CPU	COMTEN	TCS
1	25.624	23.985	3.473	0.187
2	40.210	37.017	7.803	0.452
3	56.488	51.585	11.862	0.711
4	52.650	49.129	14.506	1.056
5	51.885	48.359	10.109	0.609
6	56.855	52.563	12.764	0.830
7	56.400	49.507	11.543	0.776
8	59.259	53.039	13.985	0.967
9	35.340	31.714	7.526	0.503
10	42.862	39.485	8.805	0.549

Based on the FEDSIM study, the Customer decided not to implement the software modifications to the PACS System and no RFP was issued for the procurement of the TCS.

SUMMARY

The ECSS II Simulation model of the PACS has proven to be a useful and flexible tool to study system design alternatives. ASD has been able to use the model as part of the system development decision making process.

The model is currently being used to study two new subsystems that are being incorporated. The model is evaluating the impact of each of the subsystems prior to implementation. FEDSIM is currently performing studies of the video subsystem. A video subsystem detailed flow, the functional video scenarios, and the different types of screens comprise the basis for the model and for system design experimentation. The model is planned to be expanded to study the impact of adding the workload of new user communities. In order to perform this study, FEDSIM will model VM/370 (the operating system that is being considered).

REFERENCES

1. Sprung, James G., Beatty, P., Camery, J., and Norrell, J., "An Introduction to the Simulation of a Multiple CPU Military Communications System," Proceedings of Symposium on Simulation of Computer Systems, SIGSIM, 1976, pp. 143-150.

2. Kosy, Don, "The ECSS II Language for Simulating Computer Systems," The RAND Corporation, Santa Monica, CA, December 1975.

3. Pentagon Automated Communications System Functional Description, USACEEIA-CONUS, Fort Huachuca, AZ, March 1975.