SIZING AND ASSESSING COMPUTER DESIGN ALTERNATIVES USING SIMULATION

Gary J. Wright
Thomas L. Hannan

ABSTRACT

This paper describes the major phases of a computer system simulation study performed by the Federal Computer Performance Evaluation and Simulation Center for the Federal Aviation Administration. The Central Flow Control System, the system simulated, was in the preliminary design stage when the study was initiated. The study was undertaken to assess the performance of the proposed system, as well as the performance of system design alternatives. This study demonstrates the utility of using simulation during the early stages of a system's life cycle by providing designers insight into the system's performance and by identifying design alternatives that will improve performance.

This paper contains descriptions of the system, the simulation process and objectives, the alternatives simulated, and the results and benefits derived through the use of simulation.

I. INTRODUCTION

System simulation is a valuable tool when the object of study is complex, when the nature of its inputs is uncertain, and when the relationships among its entities may not be well-behaved. Under these conditions, formal analytical methods will not yield the desired insight into system behavior. This insight is sought for two separate activities: system analysis and system design. Analysts seek to understand the processes of existing systems so that opportunities for improvement are revealed. Designers seek to measure the behavior of alternative configurations with respect to some objectives so that an optimal selection is identified.

The simulation of the Federal Aviation Administration's (FAA) Central Flow Control (CFC) System was undertaken to answer a combination of system analysis and system design questions and the system had properties that indicated simulation would be necessary if the answers to those questions were to be found.

THE CENTRAL FLOW CONTROL SYSTEM

The FAA operates an Air Traffic Control System Command Center (ATCSCC) which has as its primary objective the balancing of national air traffic flow to minimize delays without exceeding air traffic controller capacity or jeopardizing safety. The CFC System provides automation support to enable achievement of this objective. A computer complex located in Jacksonville, Florida, is linked with the major national FAA facilities via digital computer-to-computer communications lines, and maintains the central CFC data base containing air traffic system demand and capacity data based on nation wide inputs. Demand data consists of active flight-movement data and proposed flight-plan data; capacity data consists of actual and forecast facility accommodations. The ATCSCC, which is located in FAA Headquarters in Washington, DC, has a digital communications link with the Jacksonville CFC computer complex, over which inquiries about the current and forecasted air traffic demand and capacity relationships are transmitted and responses are received. The overall CFC System is a management information system with real-time (communications), on-line (inquiry), and data base transaction aspects.

SIMULATION SCOPE AND PURPOSE

The CFC System is based on the interfacility communication capabilities of the current National Airspace System (NAS) computer hardware and the flight data processing capabilities existing at each of the 20 Air Route Traffic Control Centers (ARTCC) which make up the En Route portion of the NAS. Each ARTCC utilizes IBM-9020 (equivalent to multiprocessing IBM-360) computers with communication capabilities (ARTCC-to-ARTCC). Additionally, each ARTCC has available both demand and capacity data, as a result of its primary (air traffic control) function. The overall CFC design concept entailed the establishment of a separate IBM-9020 complex dedicated to CFC, which would tap the ARTCC's available data utilizing existing hardware and software capabilities. The primary design obstacle to be overcome was the tailoring of the IBM-9020 real-time software system to perform the Flow Control function. The simulation model was developed to analyze the behavior of this tailored CFC System in Jacksonville within the context of its communication links with the ARTCC's and the ATCSCC, and to evaluate alternative design approaches to the hardware and software configuration of the installation.

SIMULATION APPROACH AND OBJECTIVES

The Federal Computer Performance Evaluation and Simulation Center (FEDSIM) was requested to develop a high-level hardware and software simulation model of the CFC installation. The model was to utilize the pertinent representations of the NAS En Route IBM-9020 hardware and operating system already contained in a model validated and in use at the National Aviation Facilities Experimental Center (NAFEC). The initial hardware configuration was dictated when an existing ARTCC installation was decommissioned and made available as the result of a site hardware upgrade. The basic software architecture was selected by an intra-agency team for its reliability and maintenance. Previous experience indicated that the selected hardware and software basal configuration was feasible. The simulation model was developed with the essential features of this architecture assumed fixed and with certain internal features and external constraints parameterized for ease of experimentation. Prior to model development, a specific set of experiments was agreed to and the desired set of system behavioral information was requested.

II. SYSTEM DESCRIPTION

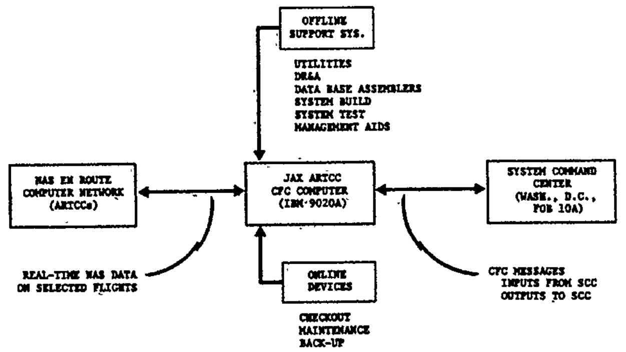
A block diagram of the CFC System is shown in Illustration 1. A more detailed description of the NAS and SCC interfaces and the CFC (Jacksonville) hardware and software configurations is presented below.

NAS (ARTCC) INTERFACE

Data from the NAS En Route system is collected by means of a store-and-forward network. The CFC computer complex is directly connected to five ARTCC's (Los Angeles, Kansas City, Cleveland, Washington, and Jacksonville), which function as store-and-forward centers. The remaining fifteen ARTCC's, with the exception of Seattle and Boston, are directly connected to the store-and-forward centers. Because of computer capacity and communications constraints, Seattle and Boston traffic is

ILLUSTRATION 1

CFC System Block Diagram



relayed via the Salt Lake City and New York ARTCC's, respectively. The five store-and-forward links are identical 2400-baud dedicated voice-grade land lines.

ATCSCC INTERFACE

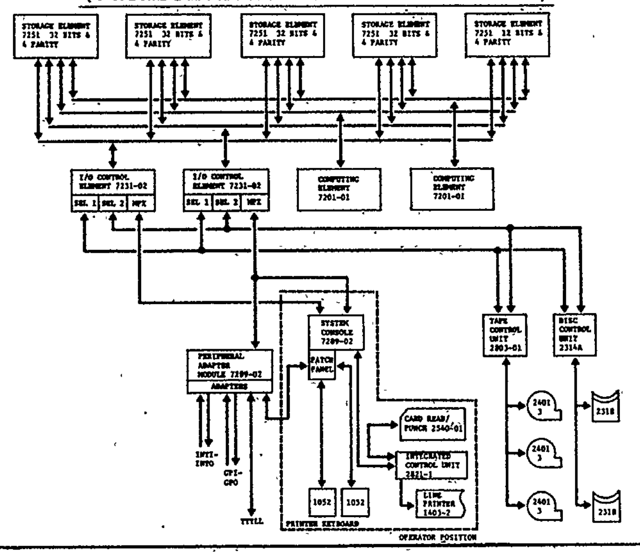
Inquiries from, and responses to, the ATCSCC are transmitted via five multiplexed, half-duplex, 1200-baud communication lines. The man-machine interface is at the data terminal equipment (DTE), which is comprised of a CRT, a keyboard, and a low-speed printer.

CFC COMPUTER COMPLEX

The CFC computer is a dual processor with 1.28 million bytes of core memory. The configuration is depicted in Illustration 2. The computer complex runs under control of the NAS Monitor (Version A3d2.4), a unique fail-safe, fail-soft (through automatic hardware reconfiguration) operating system tailored for the CFC.

ILLUSTRATION 2

Data Processing Configuration Diagram
(Jacksonville CFC Installation)



CFC APPLICATIONS PROCESSORS

The CFC System simultaneously, but asynchronously, performs two distinct functions; it maintains the currency of the CFC data base (DATA BASE UPDATE) and responds to inquiries (DATA COUNT, DATA LIST, and SIMULATION).

System stimuli are in the form of discrete, independent messages; Table 1 classifies system messages by function. Each message occurrence is allocated its own message processor copy, thus allowing for dynamic adjustment to message load composition. To prevent message load saturation at either the input or output functions, core-queue extension is provided. The NAS Monitor is augmented by a layer of executive software that provides message management and Monitor interface services. Illustration 3 shows these functions, as well as those of the application and data base software. In the initial implementation, both application and data base software are physically combined in each individual message processor; code was developed, however, in such a fashion that data base software could be made a single (probably reentrant) physical system service.

DATA BASE ACCESS METHODS

The major data base file consists of flight plans, both active and proposed. To identify uniquely a specific flight record, the fully-concatenated key AIRCRAFT-ID/DEPARTURE-AIRPORT/ARRIVAL-AIRPORT/DATE-AND-TIME-OF-DEPARTURE must be specified. To minimize the number of linkages (for both integrity and efficiency reasons) the file is physically partitioned into cells ("buckets") based on the key DEPARTURE-AIRPORT/ARRIVAL-AIRPORT. Uniquely-identified flight records are selected by retrieving the partition and searching its contents ("slots") in core. Illustration 4 shows the partition-retrieval process. In most instances, this is the access method used, because most queries are concerned with air traffic loads at some place and at some time. An additional access method, however, had to be provided for three messages (ACTV, INHB, LIFP) in which only AIRCRAFT-ID or AIRLINE-OPERATOR was specified. This access method was implemented by providing a pointer to the relative record address of each record whose AIRCRAFT-ID matched the request; the partition (relative record address divided by partition size) is retrieved, and the record (relative record address module partition size) is extracted. This process is shown in Illustration 5.

III. SIMULATION OBJECTIVES

In general, the simulation model was developed to assist CFC system designers to (1) better understand the behavior of the

TABLE 1

CFC Message Summary

DATA COUNT	DATA LIST	SIMULATION	DATA BASE UPDATE
DEMD Departure Demand	LISD List Departures	ARRD Arrival Delay Prediction	ACTV Activate Flight Plan
DEMA Arrival Demand	LISA List Arrivals	FADP Fuel Advisory Departure Block Times	INHB Inhibit Flight Plan
DESD Future Departure Demand	LIFP List Flight Plan	FADF Fuel Advisory Departure Estimated Departure Clearance Times	FPSD Add Flight Plan CXSD Delete Flight Plan CAPS Set Landing Capacities
DESA Future Arrival Demand FIXL Fix Loading	CAPL List Landing Capacities GAEL List General Aviation Estimates	FADT Fuel Advisory Departure Test QFLW Quota Flow First Tier	GAES Set General Aviation Estimates
DLDY Departure Delay Test		QFLZ Quota Flow by Zone	NAS Messages FP Flight Plan DM Departure RS Remove Strip

ILLUSTRATION 3

Message Functions

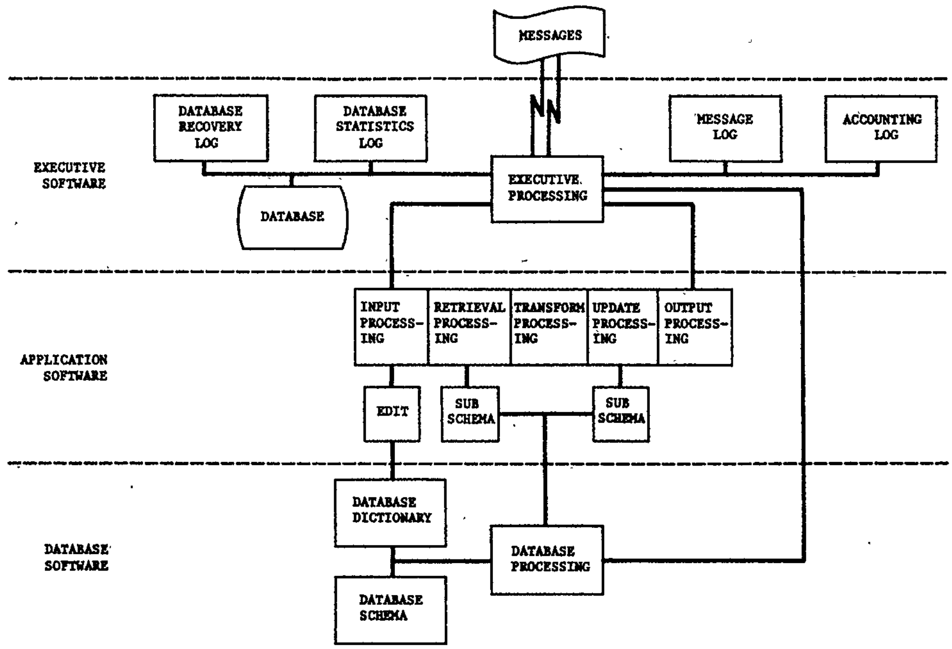


ILLUSTRATION 4

Illustration of Access by ARR/DEP Key

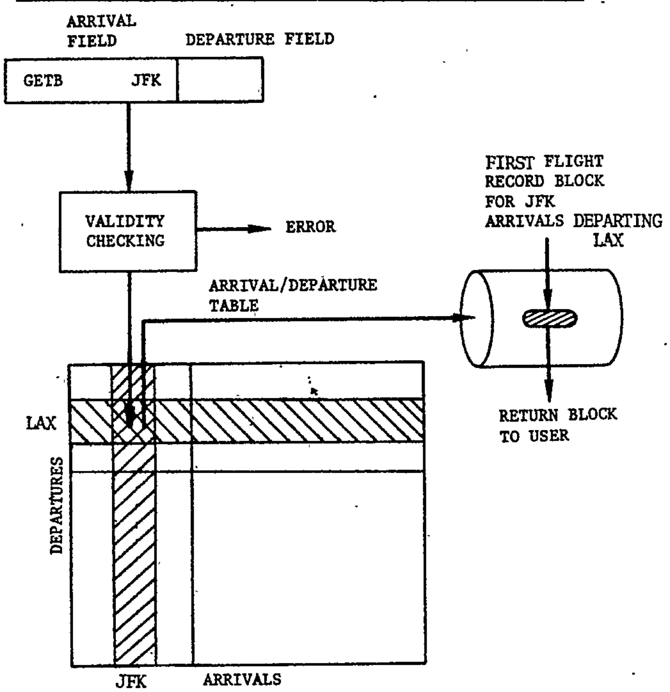
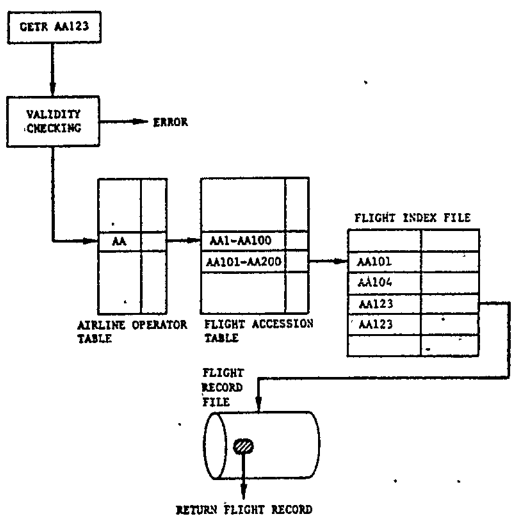


ILLUSTRATION 5

Illustration of Access by Aircraft (Flight) Ident.



system as envisioned at successive phases of the design process, so that opportunities for improvement would not be missed and (2) assess proposed design alternatives for system improvements so that the system design would remain consistent with overall system objectives.

In keeping with this strategy, a series of simulation experiments was conducted upon completion of both the preliminary design phase and the detailed design phase as part of the formal design review process (Preliminary Design Review and Critical Design Review). Specific objectives of these series of experiments are discussed below.

PERFORMANCE CRITERIA

The set of inquiry and update messages (transactions) to be serviced by the CFC System was specified by the user organization (Air Traffic Service, Federal Aviation Administration) as part of the system's requirements. In addition to the message set, Air Traffic Service also specified acceptable message response times; each message would be acknowledged with a message passed edit criteria (accept) or a message failed edit criteria (reject) response within two seconds (mean) following message entry; and each message would have completed its processing and initiated its output report within two minutes (mean) following message entry. A major objective of the simulation experiments was to measure the effect of alternative design strategies on both acknowledgement and processing termination response times.

SYSTEM RESOURCE CONSTRAINTS

Since the CFC System design was constrained by a fixed hardware configuration, a primary objective of simulation experimentation was to gather resource utilization statistics so that any problems attributable to hardware difficiencies could be discovered and subsequently alleviated (or at least not aggrevated) by software approaches.

The software architecture ultimately selected by the design team was based on the concept of multiple instances (either complete copies or reentrant modules) of independent message processors. Intuitively, the team felt that system performance would be extremely sensitive to message processor (memory) size beyond a nominal threshold. The simulation model was to be used to understand the effects of program size on system performance and to assess the expected level of system performance as program size estimates became more refined as a result of more detailed design activity.

The data base access approaches embodied in the CFC design were specifically selected to offset what was felt to be a major system hardware inadequacy--a single disk controller and channel. The simulation model was to be designed to provide a full complement of statistics regarding disk access activity so that the ramifications of data storage volumes and retrieval techniques could be fully understood.

ASSESSMENT OF ALTERNATIVES

As indicated, certain design areas were considered "soft," such as program core-use strategy and disk-access approach. The model was to be developed in such a way as to allow easy experimentation with alternate approaches in these areas.

The model's representation of the message processors had to allow the user to optionally specify serially-reusable, multiple-copy, or reentrant processors and had to provide easily-changeable core and execution-time requirements. Such system tuning features as the maximum number of message processor (by type) copies which were simultaneously allowed in core also needed to be represented so that the system performance sentitivity to their values could be evaluated.

Because disk limitation was considered significant, the model was required to accommodate expansion in this area and provide adequate statistics for cost-benefit analysis should this area become critical.

IV. SIMULATION MODEL

Based on the previous description of the CFC System and on the previously-defined simulation objectives, a simulation model was designed and developed. The model was written in the Extendable Computer System Simulator II (ECSS II) language. ECSS II, a specialized language based on SIMSCRIPT II.5, provides English-like statements to describe hardware characteristics, software operations, and resource requests.

The model simulated (at varying levels of detail) the hardware, the application software, and the operating system. The representations of the hardware and operating system were derived from the NAS Systems Model developed at NAFEC. The NAS Systems Model was written in ECSS I and simulated the major subsystems of NAS. It was designed and implemented by IBM for NAFEC's use and has been validated against hardware and software measurement data. Because of the language similarity of the two models, it was a fairly easy task to extract the necessary modules from the NAS Systems Model and to design the interfaces to the CFC executive and message processors. FEDSIM paid particular attention to the disk I/O routine, since the frequency and length of disk (data base) accesses were conjectured to be the cause of potential system bottlenecks.

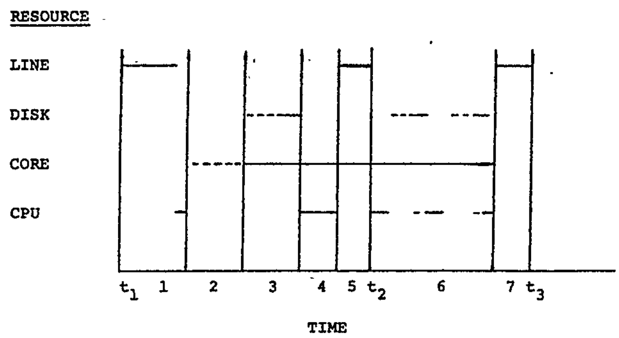
The application program sizes, timings, and behavior were obtained primarily from the FAA system designers. Because of the expected volatility of the application software characteristics, the model was designed to allow the user to change these characteristics at model execution time. Application characteristics that could be changed at execution time included the following: module type (serially-reusable, multiple-copy, or reentrant), maximum number of copies allowed concurrently in core if module type was multiple-copy, size (code and data), execution times, and frequency of message processor activity.

Other parameters that could be changed at model run time included: the starting cylinder and disk address of the data base cells, the number of data base disks and channels, the number and transmission rate of the communication lines, whether terminal responses were to be spooled to disk or directly sent to the terminals, and whether or not a data base read was required (for security reasons) prior to a data base write.

The standard ECSS II output reports include hardware queueing and utilization statistics, and timing statistics (CPU and I/O times, memory and I/O queue times, etc.) for the modeled jobs. These reports were augmented in the CFC model with response time statistics by message type, the number of data base accesses performed by each message type, the number of concurrent copies of each message processor in core, and with a log of the messages (including their parameters) simulated.

ILLUSTRATION 6

Edit Complete Resource Requests



CODE	ACTION
1	Send the message over the line and process the interrupt
2	Allocate core for the message processor
3	Fetch the message processor from disk, if necessary
4	Edit the message
5	Send the edit complete message
6	Access the data base and process the message
7	Send the message response
$t_2 - t_1$ =	Edit complete time
$t_3 - t_1$ =	Message response time

V. SIMULATION RESULTS

Before discussing the simulation statistics, a discussion of the primary performance measures, the edit complete time and response time follows. The edit complete time is delimited by the time the terminal operator pushes the send key to transmit the input message and the time the edit complete message is received. The message response time is the sum of the edit complete time, the message processing time, and the message response line transmission time.

To assist in analyzing the variability of the edit complete times and the message response times from simulation experiment to experiment, Illustration 6 has been included to depict the major resource requests that comprise these times. The illustration is not drawn to a time scale. With the exception of fetching the message processor from disk, which occurred only if the load module was not in core, the solid lines represent constant resource times among the experiments. The dashed lines represent queueing delays for the resources. The variability in these delays (codes 2 and 3) plus the frequency of program fetches caused the variability in edit complete times.

One point which may not be obvious from this illustration is that if the queueing delays for the disk and CPU can be reduced once the message begins its processing (code 6), then the core residency time (codes 3-6) for message processor will decrease. This decrease will cause a decrease in core allocation time (code 3) and, consequently, a decrease in edit complete time.

FIRST SERIES OF EXPERIMENTS

A series of three simulation experiments was performed with the CFC model near the completion of the preliminary design phase for the CFC system. The first simulation run (or baseline run) simulated the CFC System, as described earlier. The message processors were assumed to be serially reusable. The second run, Experiment 1, simulated the baseline hardware and software; but the message volumes were doubled by halving the interarrival times. The final run, Experiment 2, was identical to Experiment 1, except that the data base was split between two disks connected to separate selector channels. Each run simulated one hour and produced cumulative reports every 20 minutes.

Table 2 displays the edit complete and response time statistics by message type. All of the mean edit complete times for the message types fell within the two-second performance limit. However, DM and LISD messages violated this limit at least once. For DM, FP and RS messages, this two-second limit does not apply, since no edit complete messages are issued for these message types (all three are input from the ARTCC's). The reason LISD violated this limit was apparent from the message trace which showed that two LISD messages arrived within three seconds of one another. Because the message processors were assumed to be serially reusable, the second message was queued for its processing.

TABLE 2

Baseline Response Time Statistics

MSG TYPE	ED COMPLETE NUM	TIME MEAN	STD	RESPONSE TIME MEAN	STD
ACTV	1	0.62	0.00	190.04	0.00
ARRD	5	0.72	0.02	8.57	2.58
CAPL	27	0.50	0.05	1.55	0.13
CAPS	15	0.50	0.05	1.81	0.43
CXSD	38	0.53	0.19	2.13	0.47
DEMA	11	0.70	0.34	6.94	8.42
DEMD	4	1.04	0.56	11.95	11.29
DESA	2	0.69	0.02	9.74	6.19
DESD	1	0.70	0.00	5.32	0.00
DLDY	2	0.69	0.03	1.79	0.14
DM	612	0.29	0.22	0.95	0.41
FADF	0				
FADP	2	0.82	0.01	19.61	8.74
FADT	5	0.67	0.15	9.68	3.09
FIXL	1	0.66	0.00	3.01	0.00
FP	57	0.24	0.02	1.04	0.34
FPSD	24	0.51	0.06	2.25	0.43
GAFL	28	0.55	0.20	1.63	0.26
GAFS	16	0.58	0.28	1.97	0.45
INHB	1	0.66	0.00	501.29	0.00
LIFP	2	0.59	0.12	1.88	0.11
LISA	4	1.07	0.53	10.78	5.12
LISD	9	1.02	0.65	17.45	20.28
QFLW	0				
QFLZ	1	0.77	0.00	9.28	0.00
RS	74	0.51	0.17	0.92	0.30

All response times except those for the ACTV and INHB messages fell within the two-minute limit. The high response times for these message types (3.2 minutes for ACTV and 8.35 minutes for INHB) were caused by their excessive number of data base accesses. These two messages accounted for about 30% of the total data base accesses for all messages. Clearly, any scheme that could be designed to reduce the number of accesses for the ACTV and INHB messages would improve not only their response times, but also the response times of the other message types.

The hardware utilization statistics revealed that no resource was heavily utilized in processing the baseline message load. The data base disk had the highest utilization (29%) of all the resources. Thus, it appeared that, as the message load increased, the disk would be the limiting resource on message response time.

Table 3 shows the response time statistics from Experiment 1. Although these mean times were greater than those from the baseline, they still fell within the established performance limits. As before, the exceptions to the edit complete time of two seconds were caused by the serially-reusable load module characteristic rather than by any hardware constraint.

TABLE 3

Experiment 1 Response Time Statistics

MSG TYPE	ED COMPLETE NUM	TIME MEAN	STD	RESPONSE TIME MEAN	STD
ACTV	1	0.63	0.00	265.57	0.00
ARRD	11	0.68	0.12	17.93	10.89
CAPL	51	0.50	0.05	1.63	0.17
CAPS	22	0.50	0.05	1.87	0.29
CXSD	81	0.53	0.14	2.33	0.52
DEMA	15	0.71	0.31	6.61	3.15
DEMD	17	0.63	0.14	8.64	3.75
DESA	3	0.77	0.10	10.42	7.28
DESD	3	0.74	0.05	6.09	0.87
DLDY	3	0.62	0.10	1.79	0.06
DM	1254	0.49	0.80	1.27	1.00
FADF	1	0.87	0.00	25.14	0.00
FADP	2	0.83	0.03	29.21	13.05
FADT	5	0.70	0.15	13.52	4.24
FIXL	2	0.76	0.01	11.28	6.81
FP	121	0.29	0.22	1.35	0.60
FPSD	60	0.57	0.33	2.61	0.62
GAFL	59	0.52	0.16	1.68	0.29
GAFS	30	0.56	0.24	2.02	0.48
INHB	1	0.66	0.00	570.67	0.00
LIFP	7	0.50	0.09	2.27	0.68
LISA	9	0.68	0.10	19.76	21.03
LISD	15	2.07	4.74	25.65	25.52
QFLW	1	0.80	0.00	9.92	0.00
QFLZ	2	0.78	0.01	8.22	3.71
RS	132	0.63	0.66	1.17	0.82

Hardware utilizations were still at acceptable levels. The disk utilization increased to 47% while the average utilization for the two CPU's was only 16%. The queueing times for disk requests increased by 164% (from 28 milliseconds to 74 milliseconds) from the baseline run. These increases substantiate the conjecture that the disk will become the first resource to saturate under higher message loads. Memory capacity was never a constraint, since no requests were ever queued.

To investigate the performance of an alternate hardware configuration under high message loads, Experiment 2 simulated two data base disks connected to separate selector channels. The resulting response times are shown in Table 4. These measures reflect the reduced data base access times. The most notable response time decreases occurred for the ACTV and INHB message types. The ACTV time decreased by 37%, and the INHB time decreased by 29% from Experiment 1. Decreases in response times occurred for the other message types as well.

TABLE 4

Experiment 2 Response Time Statistics

MSG TYPE	ED NUM	COMPLETE MEAN	TIME STD	RESPONSE MEAN	TIME STD
ACTV	1	0.60	0.00	166.98	0.00
ARRD	11	0.67	0.09	11.67	5.83
CAPL	51	0.49	0.05	1.52	0.08
CAPS	22	0.49	0.04	1.66	0.16
CXSD	81	0.50	0.13	1.98	0.29
DEMA	15	0.67	0.31	4.65	1.26
DEMD	17	0.58	0.14	5.23	0.83
DESA	3	0.73	0.02	5.34	1.05
DESD	3	0.70	0.01	4.99	0.21
DLDY	3	0.60	0.08	1.66	0.10
DM	1254	0.32	0.36	0.85	0.49
FADF	1	0.79	0.00	17.19	0.00
FADP	2	0.81	0.00	27.22	9.67
FADT	5	0.68	0.13	13.55	8.60
FIXL	2	0.71	0.06	6.96	3.56
FP	121	0.25	0.07	0.91	0.28
FPSD	60	0.51	0.13	2.10	0.30
GAEL	59	0.51	0.16	1.58	0.27
GAES	30	0.53	0.23	1.75	0.30
INHB	1	0.67	0.00	405.90	0.00
LIFP	7	0.50	0.06	1.88	0.19
LISA	9	0.71	0.11	17.50	17.48
LISD	15	1.17	1.50	10.98	6.36
QFLW	1	0.73	0.00	6.66	0.00
QFL7	2	0.76	0.00	7.31	2.78
RS	132	0.45	0.15	0.79	0.22

These decreases are attributable to the lower disk queue times as they decreased from 74 milliseconds to an average (for the two disks) of 17 milliseconds. Disk seek times also decreased, since the data base size on each disk was only half that of Experiment 1.

CONCLUSIONS FROM THE FIRST SERIES OF EXPERIMENTS

The following conclusions were drawn from the above simulation results:

1. The baseline hardware and software proposed for the CFC System are able to process the expected message volumes and meet the established performance criteria.

2. The baseline system has sufficient capacity to process twice the expected message volumes and still consistently perform within the established limits. The few performance violations are caused primarily by the serially-reusable characteristic of the message processors, rather than by a hardware resource bottleneck.

3. If one considers only hardware performance, the data base disk appears to be the limiting resource; that is, as message volumes continue to increase, the data base disk will cause excessive queueing delays.

4. Splitting the data base between two disks connected to separate channels alleviates queueing for data base accesses and results in fairly balanced disk and CE utilizations.

5. Because the ACTV and INHB messages account for about 30% of the data base accesses, any decrease in the number of accesses they require will improve both their response times and those of the other message types.

SECOND SERIES OF EXPERIMENTS

A second series of simulation experiments was requested by the FAA near the completion of the detailed design phase for the CFC System. The model modifications that were made to reflect the CFC System design changes and sizing estimates included the following:

1. The message processor load module type was changed from serially-reusable to multiple-copy. That is, if a message processor was processing a message and another message of the same type arrived during the simulation, the second message was no longer unconditionally enqueued for the message processor. Instead, provided memory space existed, a second copy was fetched from disk (if necessary) and activated. To prevent memory space from being exclusively allocated to one type of message processor, the model had a parameter limiting the maximum number of message processor copies which could reside in core concurrently. Once this limit was reached and another message of the same type arrived, the message was enqueued.

2. The message processor sizes were increased to reflect the latest estimates.

3. The type of data base access performed by the ACTV and INHB message processors was changed to reflect the access shown in Illustration 5.

Three simulation experiments (new baseline, Experiment 3, and Experiment 4) were performed with the modified model. The parameter changes for Experiments 3 and 4 were identical to those for Experiments 1 and 2, except that the baseline message load was quadrupled.

Table 5 displays the edit complete and response time statistics reported by the updated model. Comparison of these times with the original baseline times (Table 2) show only slight differences. The notable exceptions are the response times for the ACTV and INHB messages which decreased by 66% and 75%, respectively. These decreases are attributed to the change in the type of data base access for these message processors.

TABLE 5

New Baseline Response Time Statistics

MSG TYPE	ED NUM	COMPLETE TIME MEAN	COMPLETE TIME STD	RESPONSE TIME MEAN	RESPONSE TIME STD
ACTV	1	0.67	0.00	64.63	0.00
ARRD	5	0.87	0.01	7.51	1.71
CAPL	27	0.48	0.05	1.49	0.09
CAPS	15	0.49	0.05	1.75	0.27
CXSD	38	0.61	0.28	2.18	0.49
DEMA	11	0.72	0.08	5.96	3.34
DEMD	4	1.12	0.62	6.67	2.08
DESA	2	0.75	0.04	4.20	0.55
DESD	1	0.72	0.00	5.30	0.00
DLDY	2	0.70	0.01	1.78	0.06
DM	612	0.14	0.06	0.78	0.30
FADF	0				
FADP	2	0.93	0.01	26.74	3.01
FADT	5	0.87	0.20	12.28	4.24
FIXL	1	0.70	0.00	2.95	0.00
FP	57	0.34	0.14	1.08	0.31
FPSD	24	0.73	0.14	2.38	0.39
GAFL	28	0.52	0.21	1.53	0.21
GAFS	16	0.58	0.29	1.89	0.37
UNHB	1	0.70	0.00	124.52	0.00
LIFP	2	0.71	0.00	2.02	0.06
LISA	4	0.65	0.12	23.93	14.58
LISD	9	0.83	0.40	7.90	1.79
QFLW	0				
QFLZ	1	0.92	0.00	9.43	0.00
RS	74	0.43	0.20	0.84	0.33

To assist in the analysis of the model's performance statistics, some composite measures are shown in Table 6. All times are expressed in seconds. Since the ARTCC messages did not receive edit complete messages or responses at processor completion, it is the ATCSCC message times that are of primary importance. The mean edit complete times for the ATCSCC messages are nearly identical in the original and new baseline runs. It appears that the elimination of any queueing for the serially-reusable message processors was offset by the increase in the number of times the programs had to be loaded from disk.

The decrease in response time (7.40 seconds to 4.59 seconds) was largely due to the

TABLE 6

Baseline, Experiment 3, Experiment 4 Composite Measures

MEASURE	BASELINE	NEW BASELINE	EXP. 3	EXP. 4
MEAN EDIT COMPLETE TIME FOR ALL MESSAGES	0.37	0.28	1.57	0.33
MEAN EDIT COMPLETE TIME FOR SCC MESSAGES	0.60	0.63	1.40	0.73
MEAN RESPONSE TIME FOR ALL MESSAGES	2.31	1.61	6.25	1.97
MEAN RESPONSE TIME FOR SCC MESSAGES	7.40	4.59	14.72	5.47
MEAN RESPONSE TIME FOR SCC MESSAGES EXCLUDING ACTV & INHB	3.96	3.68	N/A	N/A
NUMBER OF PROGRAM FETCHES FOR ALL PROGRAMS	50	150	479	229
NUMBER OF PROGRAM FETCHES FOR SCC PROGRAMS	47	84	127	114

drastic decreases in the ACTV and INHB response times. This is supported by the similar response times reported for the remaining ATCSCC messages (3.96 seconds versus 3.68 seconds).

For Experiment 3, the mean edit complete time for the ATCSCC messages more than doubled and their mean response time more than tripled. The reasons for these increases are apparent from the resource utilization and queueing delays reported in Table 7. Clearly, it is the data base disk that was saturated. Although the data base disk was not accessed during the edit processing, the large increase in message processing time (code 6 in Illustration 6) caused longer core residency times. These longer times caused the 1.7 second increase in core allocation time (code 2) which is a component of edit complete time.

TABLE 7

Resource Utilization and Queueing Statistics

MEASURE	NEW BASELINE	EXPERIMENT 3	EXPERIMENT 4
CPU UTILIZATION[1]	15.7%	65.6%	53.2%
DISK UTILIZATION	18.0%	92.6%	73.0%[2]
DISK QUEUE TIME	18 ms	371 ms	52 ms
MEMORY UTILIZATION	96.9%	96.7%	96.9%
MEMORY QUEUE TIME	0 ms	1689 ms	5 ms

[1]The sum of the utilizations for two CPU's.

[2]The sum of the utilizations for two disks.

Other resources which appeared to be unable to process the high message load were the ATCSCC input terminals. The terminals were marked busy from the time the message was sent until the complete message response was received. The average percent busy time for the terminals was over 95%. The mean time a message had to wait for a free terminal was over two minutes. Although this wait time is not a component of the edit time, it appeared to affect performance in that only 896 input messages were received in Experiment 3, whereas 942 messages were received in the new baseline. However, because of the data base disk saturation, it is doubtful that more terminals would allow more messages to be processed. It is likely that the terminal wait time would become core and disk wait times.

Another contributor to the high terminal busy times were the long output messages generated by some of the message processors (for example, the list processors). A possible design consideration might be to direct these long responses to a separate output terminal, thus freeing the input terminal for message transmissions.

The two data base disk configuration (Experiment 4) greatly improved performance. The composite measures (Table 6) show that this configuration performed nearly as well as the new baseline configuration did in processing one fourth the message load.

Once again the resource utilization and queueing times reported in Table 7 reveal why the performance measures from Experiment 4 are markedly improved over those from Experiment 3. The decrease in disk queue time enabled messages to be processed faster and core to be freed sooner which, in turn, resulted in shorter core allocation times (5 milliseconds versus 1.7 seconds). The shorter message processing times also allowed the terminals to be freed sooner as is evidenced by the drop in percent busy time (66%) for the terminals and in the decrease to 2.6 seconds in terminal wait time.

One final measure of interest is the mean number of message processor copies in core during the simulation. Table 8 contains these values for the last three simulation runs. Only those message processors whose mean exceeded 0.5 for at least one of the simulations are reported. A possible design alternative to consider is to make some number of copies core resident (for example, three DM copies in the new baseline). Such a design change would reduce the number of program fetches for these message processors which would probably cause increases to the number of program fetches for the other processors. Certainly, only the more critical message processors should be considered for this core residency option. Additionally, experimenting with the maximun number of copies allowed in core may reduce the number of program fetches.

TABLE 8

Mean Number of Message Processor Copies in Core

MESSAGE PROCESSOR	NEW BASELINE	EXPERIMENT 3	EXPERIMENT 4
CAPL	0.95	0.26	0.67
CAPS	1.00	0.17	0.65
CXSD	0.58	0.26	0.37
DM	2.11	4.22	2.78
GAEL	0.98	0.29	0.94
GAES	0.94	0.40	0.49
RS	0.87	0.72	0.76

CONCLUSIONS FROM THE SECOND SERIES OF EXPERIMENTS

The following conclusions are based on the simulation results obtained from the model.

1. The results from the model modified to reflect the latest CFC System design changes and sizing estimates indicate that the system performs well within the established criteria.

2. The change made to the type of data base access for the ACTV and INHB message processors reduced their respective response times by 65% and 75%.

3. The change in load module type for the message processors from serially-reusable to multiple-copy had little effect on the mean edit complete times. The elimination of any queueing for the serially-reusable processors appears to be offset by the increase in the number of times the programs had to be loaded from disk.

4. Quadrupling the expected message input rates caused the data base disk to become saturated. Five of the message processor types no longer meet the mean edit complete time criterion. The increased disk queueing delays caused the edit complete times to double and the response times to triple.

5. Splitting the data base between two disks connected to separate channels alleviated the queueing for the data base accesses and resulted in a system that

performed nearly as well as the baseline configuration. The configuration with two data base disks should provide sufficient capacity for processing four times the expected input message rates.

RECOMMENDATIONS

1. A CFC System design alternative that should be considered is to make a number of copies of the critical message processors core resident. This would reduce the number of program fetches for these message processors, but would probably increase the number of disk fetches for the less critical processors.

2. The maximum number of message processor copies allowed concurrently in core should be varied in subsequent simulation experiments to determine which combination of values minimizes the number of program fetches for the critical message processors.

3. Consideration should be given to directing large message responses to a separate output terminal, thus freeing the input terminal for additional message transmissions. An alternative to this hardware change would be to hold the large responses on the message spool disk until the high terminal activity subsides.

It should be noted that the FAA independently determined the need for a core residency option for the message processors and also realized the need for a medium-speed printer at the ATCSCC. Thus, the simulation results only supported their assessment.

VI. APPLICATION OF RESULTS

Results of simulation experimentation fell into three general categories; (1) statements regarding the design's ability to meet the requirements, (2) statements regarding hypotheses about the impact design alternatives had on system performance, and (3) statements regarding areas in the design that had room for improvement.

ASSESSMENT OF DESIGN ADEQUACY

The first set of simulation experiments indicated that the preliminary design could be expected to fulfill requirements with a wide margin of surplus capacity if the system could be implemented within sizing and timing estimates. The second set of simulation experiments indicated that the detailed design could also be expected to fulfill requirements with a wide margin of surplus capacity if it could be built within sizing and timing estimates. These indications constituted an integral part of the preliminary design review and critical design review process and significantly contributed to management's resolve to continue system development.

ASSESSMENT OF DESIGN ALTERNATIVES

Simulation experiments suggested that ACTV and INHB update processing was inefficient. An alternative method was designed and simulated. The simulation results indicated the new method should be incorporated into the ACTV and INHB design, and it was.

Initial results indicated that serially-reusable message processors would occasionally result in undesirable operator waits when back-to-back inquiries of the same type were received. Therefore, multiple-copy message processors were proposed, simulated, and incorporated into the system design.

In both the first and second series of experiments, the two-disk channel configuration performed significantly better than the initially-specified single-channel hardware. Because of this result, the system user is currently developing a cost-benefit evaluation of expanding system disk capabilities.

IMPROVEMENT IDENTIFICATION

The second set of experiments provided data suggesting that the system could be tuned by adjusting the maximum-processor-copy parameters and the sticking-power parameters. Results were inconclusive, but the system design was expanded to support the implementation of these "tuning" mechanisms so that experimentation could be performed on the "live" system.

A recommendation arising from the second series of experiments related to the congestion of terminal circuits due to high-volume response data when an inquiry called for extensive listings. In fact, since a medium-speed printer is available at the ATCSCC for use in such cases, it is planned that its use will be a standard operating procedure. The severity of the problem as indicated by the model results, however, has led to analysis of automatic routing of reports larger than a nominal value.

Overall, the simulation model has proven valuable in both system analysis and system design activities. In each case when the simulation results indicated that a change was clearly needed, the change was made. In each case where change was suggested as having possible value, the change was at least enabled. The use of simulation in this development effort has been of significant value; it should be of even more value in developments having more complexity, more input uncertainty, or more mathematically intractable relationships.

BIBLIOGRAPHY

1. Broglio, C. J. and Hannan, T. L., "Central Flow Control Automation - An Evolutionary Approach," Systems Research and Development Service Progress Report, FAA-RD-78-90, Washington, DC, August 1978, pp. 99-107.

2. Systems Model Design Document, ATC Systems Division, NAFEC, Atlantic City, NJ.