Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

SIMULATING THE UNIVAC 1100/82 USING UNIVAC 1100/22 CHARACTERISTICS

Melissa J. Walls
The MITRE Corporation
10210 Greenbelt Road
Seabrook, Maryland 20706

The work described in this paper was performed by The MITRE Corporation for the System Engineering Branch of the Network Control Center Division of NASA's Goddard Space Flight Center. The objective of the work was to evaluate the adequacy of a U1100/82 based computer system to support future workloads and software capabilities. Since no data on the U1100/82 system existed at the beginning of the task, a model of the U1100/82 system was extrapolated from a model of an existing U1100/22 based system with similar software. The extrapolated model was then used to perform the evaluation. This paper discusses the assumptions and techniques used to develop the extrapolated model, the model calibration and the success of the modeling approach.

1. INTRODUCTION

The National Aeronautics and Space Administration (NASA) is making many enhancements to improve the support of the space shuttle and free-flying satellites in the 1980's. A major enhancement will be the implementation of the Tracking and Data Relay Satellite System (TDRSS). TDRSS consists of three satellites and associated ground support equipment. It will provide almost full orbital coverage for tracking and data relay for the space shuttle and simultaneously up to 22 other earth orbiting spacecraft. The satellites will relay their communications to a ground station at White Sands, New Mexico. The information will be distributed from the White Sands ground terminal to several control centers and users, in particular, Goddard Space Flight Center (GSFC) in Greenbelt, MD and at Johnson Space Center (JSC) in Houston, Texas.

The Network Control Center (NCC) is a multiprocessor computer based system at GSFC that is responsible for the scheduling, resource allocation and operational control of the TDRSS Network (TN). The NCC Division at GSFC is responsible for defining the operational requirements of the NCC and for the development and installation of the NCC hardware and software. The NCC Division asked MITRE to determine if a Univac 1100/82 based NCC system could support future NCC capabilities and anticipated network loading. The approach taken was to build a simulation model of the NCC which could be executed with the anticipated workloads. Several constraints prevented a straight forward modeling approach. The most important was that the U1100/82 had not been acquired nor had the future Full Operational Capability (FOC) software development been initiated. At the beginning of the task, the only data available for model construction was collected from the existing U1100/22 system running with Interim Operational Capability (IOC) software. This IOC software provides the NCC with minimal scheduling and operational capabilities. The FOC software will provide more extensive and automated functional capabilities and will operate with a much heavier mission load.

This paper describes the method MITRE used to develop a simulation model of the U1100/82 based FOC NCC system using data from the U1100/22 based IOC system. Section 2. describes the NCC hardware and software. Section 3. describes the construction of the U1100/22 IOC model from the existing data. Section 4. describes the extrapolation of the hardware characteristics of the U1100/22 IOC model to a U1100/82 IOC model. Section 5. describes the software extrapolation from the U1100/82 IOC model to the U1100/82 FOC model. Section 6. presents comments on the modeling approach and its success.

2. NCC OVERVIEW

The NCC is a message driven system with inputs received from the network and NCC operators. The NCC consists of a communication subsytem, a display subsystem, and a main applications subsystem. This paper addresses only the third portion which is being implemented on the U1100. In the remainder of this paper "NCC" is used to refer only to the U1100 architecture in question.

The original U1100/22 based NCC hardware architecture is shown in Figure 1. It consists of

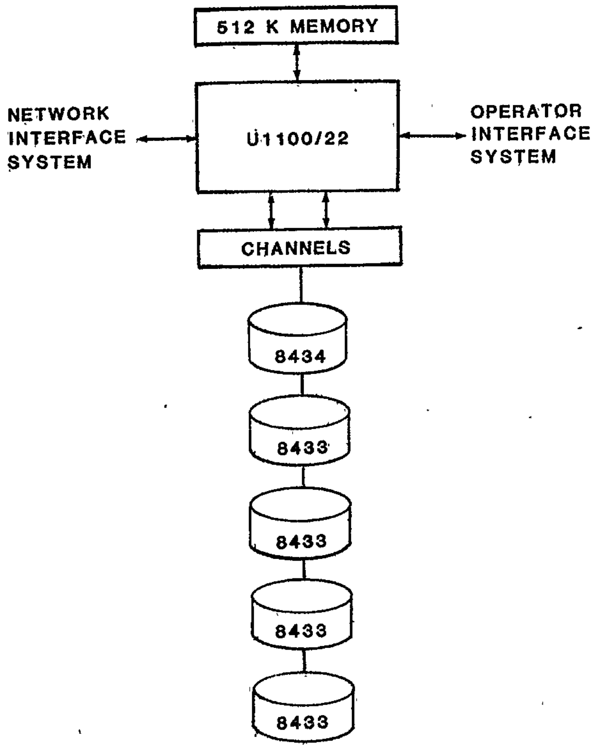

FIGURE 1: U1100/22 BASED NCC

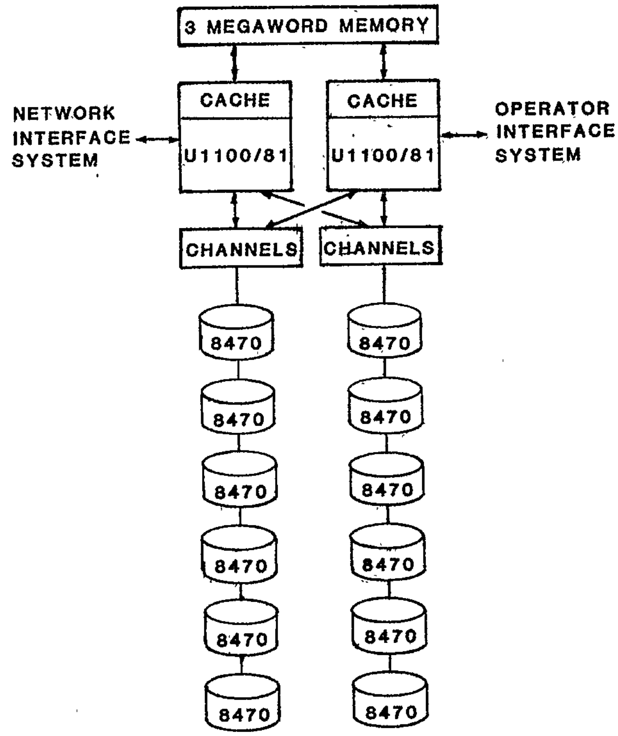

FIGURE 2: U1100/82 BASED NCC

the U1100/22 with two central processors and two Input/Output (I/O) channels for data base access. The system has 512K (K = 1024) words of memory and 5 disk drives dual ported to each CPU. The upgraded system, shown in Figure 2, consists of a Univac 1100/82 with the option of running as two U1100/81's with one processor for system development and one for NCC operations. Both configurations were analyzed in this task. The U1100/82 has 3 megawords (megaword = 1024K words) of memory, a 12K cache memory for each CPU and two strings of disk drives. Each string consists of 6 dual ported 8470 disks connected to each CPU.

The U1100/22 based NCC runs with the Univac 1100 level 33R3 operating system. The U1100/82 based system uses the level 37R2 operating system. On both systems, NCC application software is implemented as a transaction based system. Transaction is used here to imply a near real time environment in which a certain amount of time critical processing must be accomplished for each packet of data that enters the system. The system is intermediate between an interrupt driven real time system, and a batch system in which processing can be postponed arbitrarily. Transactions are processed with the Transaction Interface Package (TIP) (Sperry Univac 1976). TIP is an extension of the operating system with several special features designed to optimize execution in this environment. For example, TIP provides a mechanism for rapid loading and initiation of programs, main memory storage areas, an ability to retain programs in memory and re-execute them, mechanisms for one program to communicate with or initiate other programs, and a means of rapid access to special files.

A message entering the system triggers a sequential chain or tree of transactions. These messages enter the system either as operator inputs or as TN message traffic. There are approximately 90 different TN messages handled by the NCC. The number of different operator inputs is very large, however for the purpose of this task approximately 20 representative types of operator inputs have been incorporated into the model.

3. U1100/22 IOC MODEL DEVELOPMENT AND ANALYSIS

The largest effort in developing the initial U1100/22 IOC model was collecting and translating pertinent software and hardware architecture information into a description recognized by the modeling tool. The initial U1100/22 based IOC model and all subsequent extrapolations of the model were developed using the Performance Analyst's Workbench System (PAWS) (Information Research Associates 1982). PAWS is an event driven simulation modeling tool.

Allocatable hardware devices such as memory and channels were represented in the model by allocate nodes. The CPU and disks were represented by service nodes. The CPU was implemented as a priority quantum server and the disks with a first come first serve service (FCFS) discipline. Once the hardware architecture was translated to form the skeleton of the model, the software architecture was incorporated. Each chain of transactions triggered in response to a message entering the real system was coded into the model so that the model processes the transactions in the same order as the actual system. The amount of time a transaction spends at each node is related to the processing requirements of the transaction. The CPU, memory and word transfer requirements of each transaction were input as parameters for the appropriate nodes. The number and types of transactions active in the model during execution are dependent upon the input workload as determined by the incoming messages. In general, many transactions are active at one time and it is the competition for the memory, CPU and I/O resources that determines system performance.

Figure 3a shows the basic structure of the PAWS model. Messages from the operators and the network are generated in the model by source nodes. Arrival rates for messages are determined from observed and projected operator input rates and network workloads. The Group A nodes perform initialization operations and convert the incoming message into the first transaction in its message/transaction chain. Group B nodes determine execution priority and memory allocation and define model parameters for each transaction. If the transaction is a TIP transaction and if a load is necessary, memory is allocated to it. In the model, loading of a transaction is performed according to a probability distribution based on the transaction size. Real-time programs are always resident in memory. Group C nodes in Figure 3a constitute the CPU and I/O processing mechanism. The CPU is implemented as a quantum server with the highest priority assigned to real-time programs. A transaction must loop through the CPU until its CPU requirement is fulfilled. The I/Os are uniformly distributed among the CPU loops. Figure 3b depicts the node I/O Server in more detail. Basically, a transaction queues at a disk awaiting I/O servicing. The disk is allocated to the transaction on a FCFS basis. After the disk is allocated, the channel is allocated to the disk for an infinitesimal amount of time in order to initiate the disk seek. This time is assumed to be zero in the model. The channel is immediately deallocated after the seek is initiated. Upon completion of the seek, a channel is again allocated to the transaction to perform the transfer. Once the transfer is performed, both the channel and disk are deallocated. The transaction then starts the entire I/O process over again, returns to the CPU for continued processing, or if all its processing has been completed, moves to the next group of nodes. Group D nodes determine what happens to the transaction once its processing has been completed. The memory held by the transaction is released. If there are more transactions in the message chain, the transaction is transformed into the next transaction in the chain. The new transaction moves to Group B to obtain memory and begin processing. If the transaction is the last transaction in the chain, it is transformed into the original message that triggered it. At this point any messages to be sent out to the network or to the operators are generated. All messages terminate at sink nodes.

Partial information about the message chains was determined from draft message flow charts and through interaction with the software developers. Information on the processing requirements of each transaction was obtained from benchmark runs on the U1100/22 (Computer Sciences Corporation 1982). These benchmarks were performed with an early release of the IOC software. Each benchmark consisted of a series of sequentially increasing workloads which were divided into distinct time periods. The input workloads for each time period were clearly defined. The Univac System Instrumentation Package (SIP) (Sperry Univac 1981) and the Log Analysis Statistics Summary and Other (LASSO) (Sperry Univac 1980) programs were used in the collection and reduction of the benchmark data. SIP and LASSO data were reduced by MITRE to determine the performance characteristics of each transaction executed during the benchmark. These included values for the mean CPU time, transaction size, number of I/O's and I/O transfer sizes. These values were used to parameterize the model and to provide service rates and resource requirements at the appropriate nodes in the model.

3.1 U1100/22 IOC Model Results and Analysis

After the U1100/22 model was constructed and parameterized with the transaction performance data, it was executed with workloads identical to those used in the benchmark tests. Table 1 summarizes these workloads. The basic message types appear in the left hand column of Table 1. Input message rates are entered in the matrix for each test. Type 1 messages represent the periodic receipt of operations data by the NCC from the network. Type 2 messages represent the display of the incoming data to the operators. Type 3 messages represent the forwarding of the relevant operations data to the network users. Type 4 messages represent a user request to add an event to the schedule. These four message types represent the heaviest system load. Other message types were included in other portions of the benchmark. These workloads were not used since detailed processing information for these messages was not available. The model was only calibrated to the workloads in Table 1. The ability to use the tests, which included other message types in the model calibration, would have increased the confidence in the model when using other message types.

Table 2 compares the results of the U1100/22 model with the benchmark results. The lines labeled "benchmark" represent the observed benchmark results. The lines titled "uncalibrated" reflect the initial results of the model parameterized as described above. The CPU utilization values reported by the uncalibrated model were relatively consistent for tests B,C,D and E with an average error of 13%. Test A, representing the lightest load, showed approximately a 30% error for CPU utilization. Memory utilizations reported by the uncalibrated model were very close to the benchmark results. Definite patterns relating the results to the workload could be detected. Channel utilization began to be overestimated in tests D and E, and appeared to be related to the number of schedule requests processed during the test period. Examination of the uncalibrated results of tests A,B and C indicated that the CPU and I/O resource demands, attributed to the receipt and display of operations data, were underestimated. A review of the transaction parameterization data used in the model was performed. Subsequent analysis of the data reduced from the LASSO reports revealed some insight into the problem. Since TIP may re-execute a transaction without re-initializing it, the CPU processing value reported by LASSO was often the sum of processing for several executions of the same transaction. It was originally

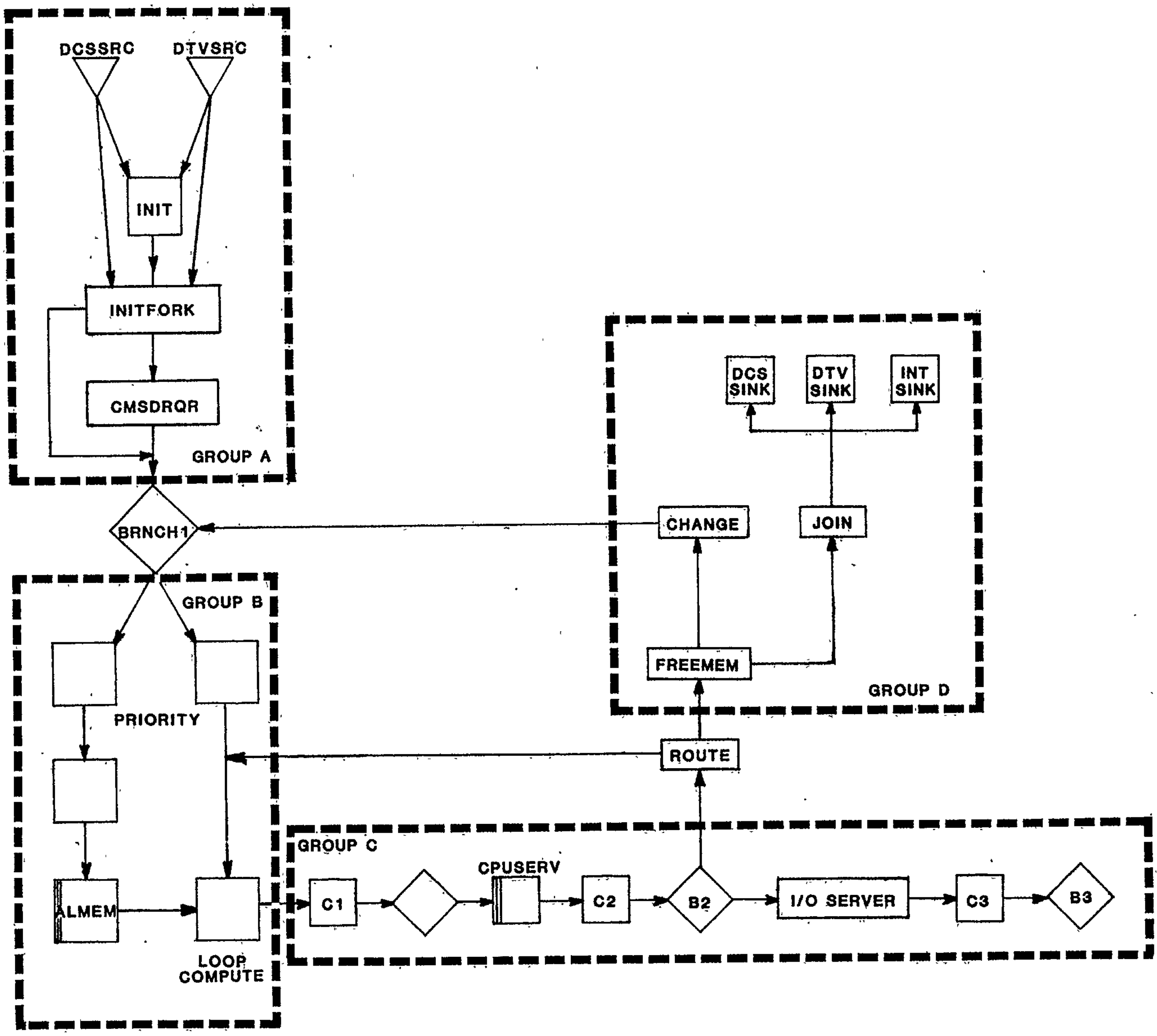

FIGURE 3A: PAWS NCC MODEL

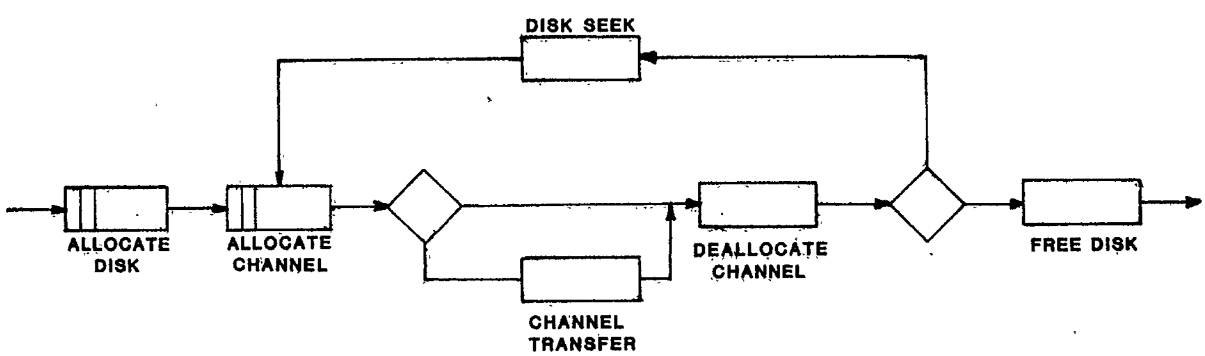

FIGURE 3B: I/O SERVER

	RATES				
MESSAGE TYPE	A	B	C	D	E
1 Receive Operations data	3/5 sec	3/5 sec	3/5 sec	3/5 sec	3/5 sec
2 Display Operations data	___	1/5 sec	1/5 sec	1/5 sec	4/5 sec
3 Send Operations data	___	___	4/5 sec	4/5 sec	4/5 sec
4 Schedule Request	___	___	___	1/min	1/min

Table 1: IOC Benchmark Workloads A-E

TEST		CPU	(ERROR)	CHANNEL	(ERROR)	MEMORY	(ERROR)
	Benchmark	6.85%		4.45%		67.7%	
A	Uncalibrated	4.75	(30.7%)	3.90	(12.7%)	69.5	(2.6%)
	Calibrated	6.89	(0.58)	4.37	(2.0)	71.0	(4.9)
	Benchmark	14.35		10.35		71.3	
B	Uncalibrated	12.37	(13.8)	6.75	(34.9)	70.9	(0.6)
	Calibrated	14.52	(1.2)	9.49	(8.3)	73.4	(3.0)
	Benchmark	23.25		16.0		72.2	
C	Uncalibrated	20.26	(12.9)	12.82	(20.0)	71.9	(0.4)
	Calibrated	23.22	(0.1)	15.97	(0.2)	75.0	(3.9)
	Benchmark	28.45		21.2		76.6	
D	Uncalibrated	31.94	(12.3)	22.75	(7.3)	78.2	(2.1)
	Calibrated	28.82	(0.3)	21.59	(1.8)	79.9	(4.3)
	Benchmark	59.25		36.25		90.2	
E	Uncalibrated	66.91	(12.9)	44.11	(21.7)	91.1	(1.0)
	Calibrated	56.40	(4.8)	37.50	(3.5)	93.8	(4.0)

Table 2: U1100/22 IOC Model Utilizations

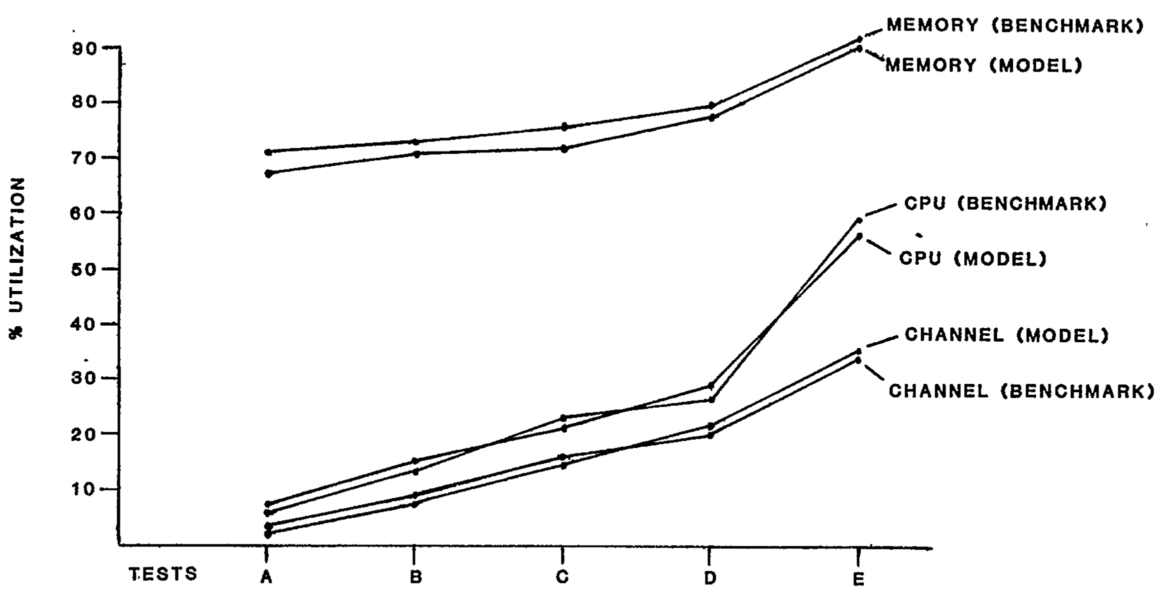

FIGURE 4: CALIBRATED U1100/22 MODEL UTILIZATIONS

assumed that a constant number of executions was summed into each reported value. After review of the data, it was determined that the number of executions summed into the reported value varied and was often less than the number originally thought to occur. On an average, fewer transaction executions were summed into the value which increased the amount of processing performed by each transaction.

Examination of the schedule processing indicated that the data used in the model represented an accepted schedule request. Accepted schedule requests generally require more processing than rejected ones. The model did not distinguish between accepted and rejected schedule requests. The data used to parameterize the model represented only an accepted request. Since a mix of accepted and rejected requests occurred in the benchmark, the model was overestimating average processing time. Examination of the benchmark results indicated that on the average a schedule request required 25% less CPU processing than the amount indicated in the model.

The model was run with the calibrated schedule request and operation data processing values. The calibrated model results are labeled in Table 2 and are shown in graph form in Figure 4. It can be observed that the average CPU utilization error was 1.6%, average channel utilization error was 3.16% and the average memory utilization error was 4.0%. These low errors indicated excellent calibration between the model and the benchmark workloads. Figure 4 depicts this calibration. The lack of additional benchmark data prevented any further validation of the model.

4. U1100/22 TO U1100/82 IOC MODEL CONVERSION

Construction of the U1100/82 IOC model was based on the U1100/22 model. The differences in hardware and operating system characteristics between the two configurations were determined and represented in the model by simple parameter changes. The U1100/22 model was expanded to include six Univac 8470 disk drives as opposed to four Univac 8433 disk drives and one Univac 8434 disk drive. Only one of the disk strings in the U1100/82 configuration was modeled since only one string is to be used during FOC operations. Four I/O channels were modeled instead of two and the disk transfer rates were changed to represent the 466,666 words per second (w/s) transfer rate of the 8470 disks, as opposed to 179,111 w/s for the 8433 disks and 279,334 w/s on the 8434 disks. The amount of memory reserved for system functions was assumed to be 364K on the U1100/22. The same amount was assumed to be reserved on the U1100/82. An additional 1 megaword was reserved for off-line processing, leaving approximately 1684K words available for application processing on the U1100/82 as opposed to 150K words on the U1100/22. The 12K cache memory of the U1100/82 was not explicitly modeled. Instead, it was incorporated along with the difference in CPU processing speeds and changes in operating system efficiency that resulted in a reduction in the effective amount of CPU processing required per transaction. It was determined from vendor literature and confirmed by the software developers that the U1100/82 processor was approximately 2.4 times faster than a U1100/22 processor. This factor was introduced into the model by dividing the CPU requirement of each transaction in the U1100/22 model by 2.4.

4.1 U1100/82 IOC Model Results and Analysis

By the time the conversion from the U1100/22 to the U1100/82 model was completed, the U1100/82 had been leased and was operational at GSFC. Benchmarks that had been run on the U1100/22 were repeated on the U1100/82. Limited performance data was available from these tests, however, the level 37 operating system does not log TIP transactions. Therefore, the characteristics of the individual TIP transactions could not be collected and compared to the data used to parameterize the extrapolated model. The only data available was a U1100/82 benchmark (Computer Sciences Corporation 1982b) with the workloads listed in Table 1. Table 3 shows the results of the U1100/82 model run with these workloads. The lines labeled "2.4" represent the initial 2.4 CPU conversion factor results. Tests B,C,D and E overestimated CPU utilization. CPU utilization was underestimated in test A. Channel utilizations could not be determined from the CSC reports and were therefore not reported, however, the table shows the predicted channel transfer utilization results from the model. Memory utilization was predicted accurately by the model and is not an area of concern.

The major errors in the model occurred in CPU prediction. The model greatly underestimated the CPU utilization of test A while tests B,C,D and E were slightly overestimated. Several software improvements had occurred between the time the model parameterization data was collected on the U1100/22 and the U1100/82 system benchmarks were performed. The data used to parameterize the model did not reflect these improvements. In particular, major changes were made to the I/O handler and message routing routines. Each message entering or leaving the NCC was affected by this change. The model did not reflect these software improvements and therefore overestimated the amount of processing attributed to each message. In order to compensate for software improvements, the CPU factor was increased to 2.7 representing a 12% improvement in software and operating system efficiency. This value was determined from analysis of the benchmark results.

The results of the model executed with the 2.7 CPU factor are also shown in Table 3. Figure 5 compares the calibrated results with the benchmark in graph form. The model underestimated the CPU utilization of test A. This test was ignored due to the extremely light workload which leaves much room for error in the results. With the exception of test D, the CPU utilizations reported for the remaining tests were within 6%. Memory utilization predictions for all workloads continued to be very accurate. These results suggested that the model should not be used to provide precise predictions of CPU utilization in extremely light loads, however the model can be used to analyze intermediate and heavy loads with acceptable accuracy. The fact that the model overestimates CPU utilization on moderate and heavy loads provides an added degree of conservatism to the evaluation of the results. Due to the lack of data, additional calibrations or validations could not be performed. The compliance of the model results with the benchmark

TEST		CPU (ERROR)	CHANNEL	MEMORY (ERROR)
A	BENCHMARK	3.4%		16.8%
	2.4	2.0 (41.2%)	2.8%	17.9 (7.4%)
	2.7	1.9 (43.8)	2.8	17.9 (7.4)
B	BENCHMARK	5.1		17.5
	2.4	5.4 (5.3)	4.8	18.3 (4.3)
	2.7	4.8 (6.3)	4.8	18.2 (4.2)
C	BENCHMARK	6.8		18.5
	2.4	8.4 (23.0)	9.5	18.4 (0.3)
	2.7	7.5 (9.4)	9.9	18.4 (0.4)
D	BENCHMARK	9.3		19.5
	2.4	12.8 (37.7)	14.5	19.5 (0.0)
	2.7	11.3 (22.0)	14.5	18.4 (5.4)
E	BENCHMARK	25.3		21.2
	2.4	27.3 (8.0)	25.89	21.5 (1.6)
	2.7	24.3 (4.3)	25.92	21.4 (1.0)

Table 3: U1100/82 IOC Model Results

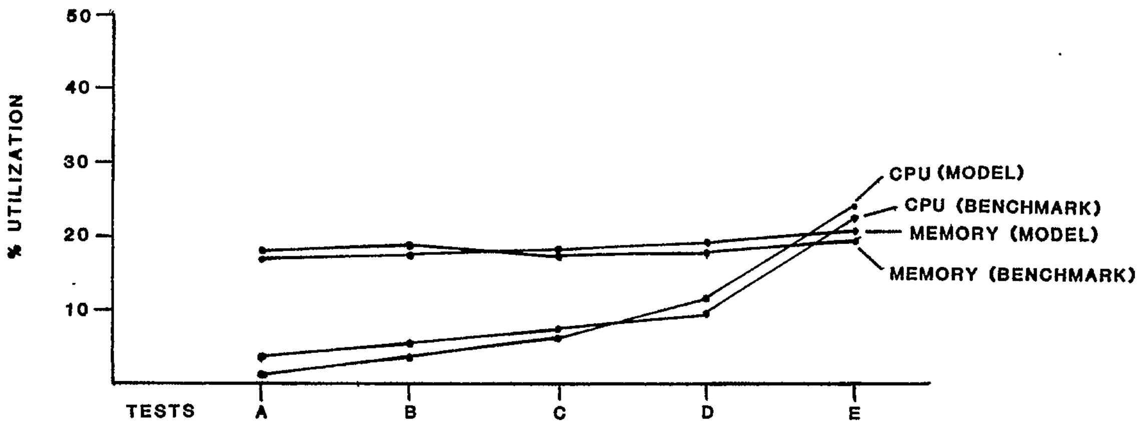

FIGURE 5: CALIBRATED U1100/82 IOC MODEL RESULTS

in this case suggests that the model can be used with confidence even though a proper calibration and validation were not performed. The message types in the workloads used to calibrate the model constitute the major load of network processing.

4.2 U1100/81 IOC Model Results

The U1100/82 IOC model was converted to a U1100/81 model by reducing the number of channels to two and the number of CPUs to one. The only U1100/81 data available for comparison to the model was from the U1100/81 benchmark (Computer Sciences Corporation 1982c) performed with a later software release than that used to develop the model. The model was used "as is" for comparison. Due to the differences in releases, however, the model should have been re-parameterized and calibrated to compensate for the differences between the releases. Due to a lack of the proper LASSO data, this was not possible.

Table 4 shows the results of the U1100/81 model parameterized with the early data and executed with the workloads used in the U1100/81 benchmark. These results are labeled "calibrated (early release)". The workloads are described in Table 5. Message types 1 through 4 are the same as before. Message type 5 represents a user request to reconfigure the ground equipment. Message type 6 is the transmission of data to support TDRSS antenna pointings. The U1100/81 model overestimated CPU utilization and underestimated memory utilization. This was attributed to the changes between the releases, primarily the improvements made in schedule processing and the number of schedule requests that were rejected. The significance of the mix of accepted and rejected schedule requests was discussed in Section 3.1 above. In the U1100/81 benchmark, approximately 50% of the schedule requests were rejected. This further reduced the average CPU processing per request. The model did not compensate for this. Examination of the results of two workloads that differed only by the processing of schedule requests, allowed a determination of the average amount of CPU time required to process a request in this benchmark. The difference in CPU utilization between the two tests was attributed to schedule request processing. It was determined that the CPU processing required by a request in the model, parameterized with the early software release data, was double the value from the benchmark. Appropriate transactions in the schedule request message/transaction chain were adjusted to reflect the lower CPU requirement in the later release. While other possible explanations for the overestimation existed, the data needed to explore any of the potential problems and concerns or to extrapolate the model to correspond to the later software release did not exist.

The model was run with the modified schedule request processing value. The results are listed in Table 4 as "recalibrated (early release)" and are shown in graph form in Figure 6. Although the model still overestimated CPU utilization, there was a considerable improvement in the prediction ability. The model underestimated memory, indicating that increases in software size occurred between the two releases. The model can be used to predict U1100/81 performance, however it must be realized that the model is based on earlier software release data. In order to accurately calibrate to the U1100/81 benchmark, more data on the later release is needed.

5. FOC MODEL DEVELOPMENT AND ANALYSIS

The U1100/82 IOC model provided a basis for extrapolation to the FOC model. Conversion of the PAWS model from IOC to FOC was based on a comparison of the FOC and IOC requirements. Those functions estimated to have significantly different processing requirements in FOC were identified and incorporated into the U1100/82 IOC model. Estimates of the expanded FOC capabilities in terms of CPU, I/O and memory requirements were determined in conjunction with personnel familiar with the IOC and FOC systems. The assumptions

		RATES		
	MESSAGE TYPE	F	G	H
1	Receive Operations data	1/sec	1/sec	1/sec
2	Display Operations data	6/5 sec	6/5 sec	6/5 sec
3	Send Operations data	1/sec	1/sec	1/sec
4	Schedule Request		1/25 sec	1/30 sec
5	Ground Configuration		1/30 sec	1/30 sec
6	Antenna Positioning			100/ min

Table 4: U1100/81 Benchmark Workloads

	TEST		CPU	CHANNEL	MEMORY
	BENCHMARK	(later release)	36.8 %	----	23.4 %
F	CALIBRATED	(early release)	44.5	7.9 %	19.8
	RECALIBRATED	(early release)	44.5	7.9	19.8
	BENCHMARK	(later release)	47.3	----	32.1
G	CALIBRATED	(early release)	80.5	9.4	26.6
	RECALIBRATED	(early release)	68.9	9.4	23.5
	BENCHMARK	(later release)	49.8	----	35.5
H	CALIBRATED	(early release)	96.5	14.6	56.5
	RECALIBRATED	(early release)	69.3	9.4	23.5

Table 5: U1100/81 IOC Model Utilizations

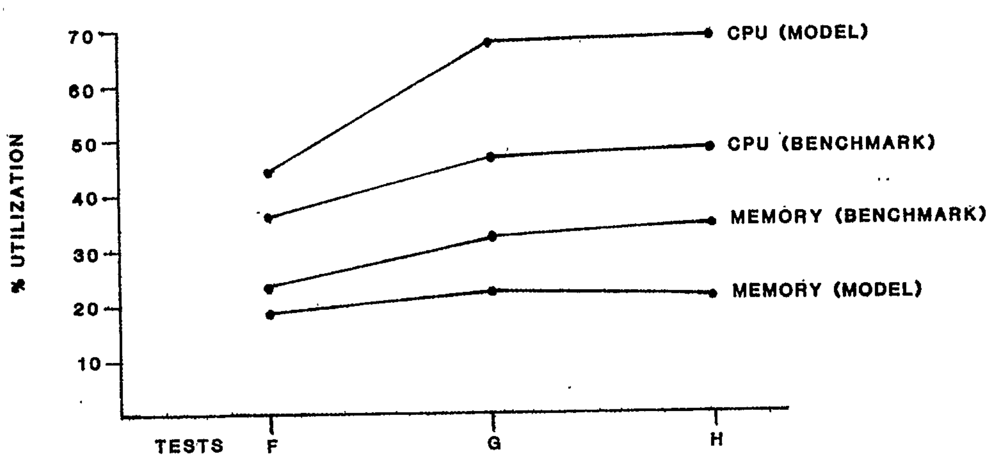

FIGURE 6: U1100/81 IOC MODEL RESULTS

used in the conversion were reviewed by NASA.

Upon completion of the conversion, workloads characteristic of the FOC timeframe were developed and executed with the FOC model. The FOC model was run with the workloads of Table 1 for comparison to the U1100/82 IOC model to assess the effects of the software extrapolation. These results are summarized in Table 6. The CPU, memory and channel utilizations were slightly higher than those experienced by the U1100/82 IOC model. This was as expected due to the increased demand for CPU, memory and channel resources due to the expanded capabilities of the scheduling, acquisition data processing and operation data functions. The FOC model results indicated that the U1100/82 could adequately support workloads A-E. Table 6 also shows the results of the U1100/81 FOC model run with the same workloads. These results indicate that the U1100/81 is also capable of supporting workloads in Table 1.

TEST		CPU	CHANNEL	MEMORY
A	U1100/82	1.865%	0.1243%	17.99%
	U1100/81	3.73	0.2885	18.0
B	U1100/82	4.805	0.5975	18.25
	U1100/81	9.61	1.1950	18.25
C	U1100/82	7.53	1.413	18.42
	U1100/81	15.06	2.825	18.45
D	U1100/82	12.54	1.59	21.57
	U1100/81	25.08	3.18	19.65
E	U1100/82	26.53	3.20	21.57
	U1100/81	52.96	6.40	22.08

Table 6: U1100/82 and U1100/81 FOC Model Utitizations

The FOC model was executed with the heavier workloads expected to occur in the FOC timeframe. The model results indicated that the U1100/82 was capable of supporting these workloads. The results of the U1100/81 FOC model indicated that the system was unstable in extreme loading conditions but could handle moderate workloads with degraded message response times. This information was NASA's main validation that the U1100/82 is capable of supporting the NCC FOC. Other applications of the model determined if the NCC, as it is currently configured for IOC, could support FOC workloads. The model results indicated that the IOC system could support early FOC workloads. This information is being used by NASA in its determination of the FOC implementation schedule.

6. COMMENTS

The method of extrapolation of the U1100/82 FOC model from the U1100/22 IOC model was very successful. The success depended mostly on the accuracy of the initial U1100/22 IOC model. This U1100/22 model was accurately calibrated with available benchmark data run with an early software release. The message types for which the model was calibrated account for the majority of processing expected to occur in FOC. As a result, the lack of calibration data for additional message types did not severely affect the modeling success. Extrapolation of the U1100/22 IOC model to the U1100/82 IOC model was relatively straight forward with the exception of the evaluation of the increase in operating system efficiency. Unfortunately, the operating system with which the U1100/82 was configured did not log TIP transactions. Therefore, no performance data on the effect of the increased CPU speed and operating system efficiency on the individual TIP transactions was available. As a result, the operating system characteristics were incorporated in the model in terms of a faster CPU speed. Calibration to the observed U1100/82 benchmark data demonstrated the success of this approach.

Development of the U1100/81 IOC model from the U1100/82 IOC model involved the change of model parameters representing memory size and the number of CPUs. Unfortunately, the actual benchmark performed on the U1100/81 system was performed with a later software release than that used to construct the model. In general, the largest errors occurred when comparing the U1100/81 model results to the benchmark results. Insufficient data was available to extrapolate the entire U1100/81 model to the later software release. However, enough data was available to recalibrate the schedule request message which has the largest processing requirements of all the messages. Recalibration of this message alone greatly improved the U1100/81 model accuracy.

Extrapolation of the U1100/82 IOC to FOC model proved to be more subjective than the hardware extrapolation of the U1100/22 to U1100/82 IOC model. The functional differences between the IOC and FOC software were assessed and expressed in the model in terms of resource requirements. No data existed with which to calibrate or validate the FOC model. Since the calibration of the U1100/82 IOC model was successful and the FOC model was extrapolated from the IOC model, it was felt that the FOC model results could be used with confidence. However, the results must be analyzed in view of the assumptions made about the FOC software. To date, the model has proven useful to NASA in its evaluations of system performance and in forecasting future needs. It is anticipated that the model will be more useful as more data is received and the model is refined. Final assessment of the model extrapolation can only be made after the FOC system is implemented.

REFERENCES

Computer Sciences Corporation (1982), Release 3.4 System Performance Summary from Data Collected at NCCDS During System Benchmark Tests on the U1100/22, NCC Performance Report No. 4, August 31, 1982.

Computer Sciences Corporation (1982b), Release 3.4 System Performance Summary from Data Collected at NCCDS During System Benchmark Tests on the U1100/82, NCC Performance Report No. 6, November 5, 1982.

Computer Sciences Corporation (1982c), Release 3.6 R1 System Performance Summary from Data Collected at NCCDS During System Benchmark Tests on the U1100/81, NCC Performance Report No. 7, February 17, 1983.

Information Research Associates (1982), PAWS 2.0, Performance Analyst's Workbench System, User's Manual, December 2, 1982.

Sperry Univac (1976), Sperry Univac 1100/80 System, Hardware System Description, UP-8351, 1976.

Sperry Univac (1980), LASSO Log Analysis Program Documentation, Goddard Space Flight Center U1100/82 System Documentation, July 30, 1980.

Sperry Univac (1981), Univac Software Instrumentation Package, Goddard Space Flight Center U1100/82 System Documentation, November, 1981.