The actual effect on the gross measures of the system were negligible.

5.2.3 Summary

This experiment made no attempt to reduce the number of punts resulting from some running job doing an RFL up after the job scheduler issued its decisions. It would seem that the misordering of 1RJ executions and the resulting level of punts has little effect on system performance. As so little effect was noted, only one delay interval, that of 50 milleseconds, was tried. Little point could be seen in pursuing this experiment further.

5.3 Experiment C - CIO-2RD-2WD Overlay

A CIO request generally results in the servicing of two LPP functions by CPM, one for CIO and one for the overlay it calls to process the I/O request. The two overlays most frequently called by CIO are 2RD and 2WD. If 2RD and 2WD were a part of CIO, each request for CIO which would require the execution of 2RD and 2WD would require one less LPP function, saving one millesecond of CPM overhead. Further, the memory of a PPU is of more than adequate size to hold the code for all three overlays at a time.

5.3.1 Experiment Description

In the model 2RD and 2WD were still represented by their individual SPG's as processes other than CIO do invoke them from time to time. They were also incorporated into the SPG for CIO.

5.3.2 Comparison of Results

Table 5.4 presents the amount of CPM overhead eliminated by this restructuring of the SPG for CIO. In both experiments, the amount of time saved is relatively small, but does result in some improvement in system performance as illustrated by Table 5.5. CPM utilization of the CPU decreased slightly while the utilization by control points increased in both models. For SIM 1 the reduction in elapsed time is roughly twice the saving in CPM time directly related to the modification. For SIM 2 the reduction is one and a half times as great.

MEASURE	SIM 1	EXP. C-1	SIM 2	EXP. C-2
Number of Times 2RD and 2WD Executed for CIO	13119	13422	15756	15678
Mean CPM Service Time for LPP's	.932 ms	.932 ms	1.018 ms	1.018 ms
CPM Time Spent Servicing LPP's from CIO for 2RD and 2WD (seconds)	12.227	0	16.039	0
CPM Time Saved by Having 2RD, 2WD, & CIO as One Overlay (seconds)	0	12.509	0	15.960

Experiment C - CIO-2RD-2WD Overlay

Table 5.4

CPM Time Saved

MEASURE	SIM 1	EXP. C-1	SIM 2	EXP. C-2
Elapsed Time (seconds)	1541.406	1514.793	1238.074	1215.130
Utilization				
CPU				
Control Points	.3959	.4029	.6788	.6916
CPM	.2620	.2516	.2506	.2423
PPU	.9955	.9943	.8772	.8693
Central Memory	.90	.89	.91	.90
Degree of Multiprogramming	12.7	12.5	9.3	9.4
Disk Channel Requests per second	14.56	14.59	16.70	16.46

Experiment C - CIO-2WD-2RD Overlay

Table 5.5

Comparison of Performance Measures

5.3.3 Summary

Combining CIO, 2RD and 2WD results in some performance improvement. Many overlays are quite small and other combinations might be considered. As the improvement was slight and CIO represents such a large fraction of PP requests, little further improvement, however, seems likely.

5.4 Experiment D - All PP Overlays on ECS

In the UT-2 system, the code for often used peripheral processes resides in central memory and is called the resident peripheral process library (RPL). The code for less often used overlays resides on the system disk. When an LPP function is processed, CPM returns the core or disk address to the PPU as appropriate. The PPU either loads the code from central memory or requests the system channel to load from disk.

There are two areas of overhead to be considered in this approach:

(1) the RPL occupies 12000_8 words of central memory; and

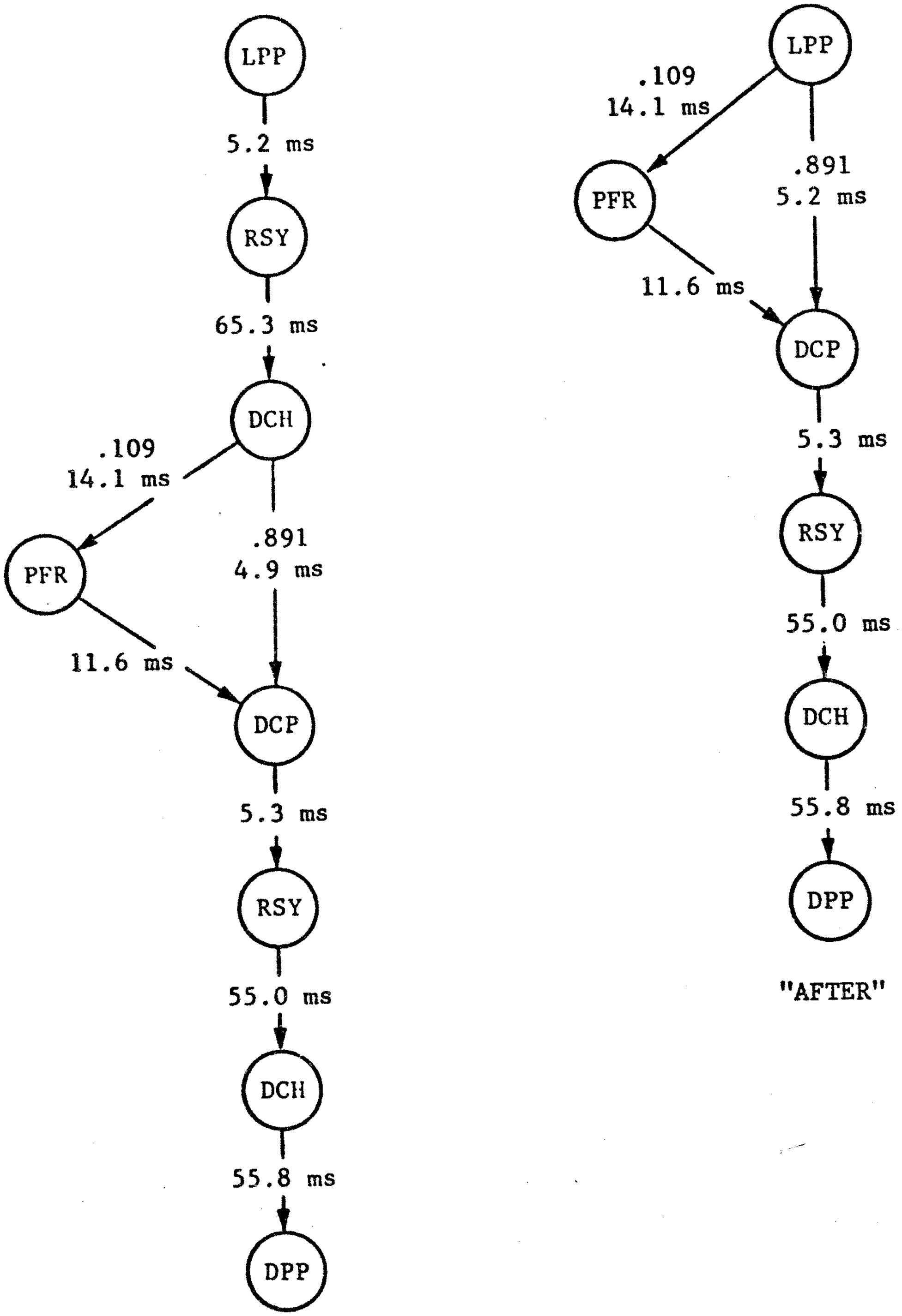
(2) a PPU loading from system disk must wait for the system channel and then spend on the order of 30 to 70 milleseconds loading the overlay.

5.4.1 Experiment Description

In this experiment all overlay code is considered to reside in ECS. The processing of an LPP function now results in CPM loading the designated code from ECS into a buffer in central memory and returning the buffer address to the PPU. The rather pessimistic overhead of one millesecond was attached to the loading from ECS.

10000_8 words of the RPL were returned to the pool of allocatable central memory. The other 2000_8 words were reserved for buffers for ECS - central memory - PPU memory transfers.

The SPG's for those processes residing on system disk had to be modified. Figure 5.4 demonstrates the typical "before" and "after" modification SPG.



"AFTER"

"BEFORE"

Figure 5.4
LOD SPG

5.4.2 Comparison of Results

The extra CPM overhead associated with loading the overlays from ECS caused the performance of SIM 2, with its already high CPU utilization, to degrade sharply as can be seen in Table 5.6.

MEASURE	SIM 1	EXP. D-1	SIM 2	EXP. D-2
Elapsed Time (seconds)	1541.406	1494.817	1238.074	1296.217
Utilization				
CPU				
Control Points	.3959	.4083	.6788	.6484
CPM	.2620	.3289	.2506	.3142
PPU	.9955	.9933	.8772	.8438
Degree of Multiprogramming	12.7	13.0	9.3	9.7
Storage Moves				
Count	13763	13911	8369	8663
Rate (per second)	8.9	9.3	6.7	6.7
Job Scheduler				
Count	4629	4347	3660	3875
Rate	3.0	2.9	2.9	3.0
Punts				
Count	702	730	516	514
Rate	.455	.4880	.417	.3965

Experiment D - All Overlays in ECS

Table 5.6

Comparison of Performance Measures

The extra overhead is absorbed by the large idle fraction in SIM 1 and some improvement in performance was achieved. The PPU time saved due to the modification was the probable cause of this improvement. As the PPU's were so saturated, this freed time was not, however, reflected in a decrease in PPU utilization.

The extra allocatable central memory resulted in a slightly higher degree of multiprogramming. However, the scheduler and storage move statistics showed little change.

Table 5.7 shows a comparison of the channel utilization statistics.

MEASURE	SIM 1	EXP. D-1	SIM 2	EXP. D-2
RCH				
Count	13715	13809	14311	14332
Mean Waiting Time	36.02	41.26	26.22	27.19
Mean Holding Time	100.52	101.9	76.18	75.54
Utilization	.2981	.3139	.2936	.2943
Balance	.7362	.7118	.7439	.7353
RCS				
Count	3076	3292	1906	1964
Mean Waiting Time	138.22	148.12	87.06	72.67
Mean Holding Time	160.84	162.34	99.45	97.4
RSY				
Count	5652	2441	4556	1880
Mean Waiting Time	148.09	146.03	88.76	90.45
Mean Holding Time	112.35	191.47	104.56	179.80
RSY + RCS				
Count	8728	5733	6462	3844
Mean Waiting Time	144.6179	147.23	88.26	81.37
Mean Holding Time	129.44	166.23	103.05	137.70
Utilization	.7329	.6375	.5379	.4316
Balance	.4723	.5303	.5386	.6285

(Time in Milleseconds)

Experiment D - All Overlays in ECS

Table 5.7

Channel Statistic Comparison

The mean waiting time does not exactly equal the service time for the corresponding functions. It does not include CPM noise that might occur. As RCS's and RSY's refer to the same disk channel, utilization and balance statistics were not computed for them individually. Although the number of RSY functions were reduced by more than half, the channel utilization was not reduced as much as one might expect. The RSY's eliminated were for requests, which, when granted, resulted in short holding times as the peripheral processes transferred were quite small. The effect on channel utilization was also decreased by an increase in RCS functions, particularly for SIM 1.

5.4.3 Summary

The extra allocatable memory seemed to have little effect. The overall effect of moving the overlays to ECS had mixed results. It decreased performance when the system was fairly balanced but improved it when the system was near thrashing.

5.5 Experiment E - Overlays on System Disk Moved to ECS

As the extra allocatable memory seemed to have little effect on system performance in the preceding experiment, a modified approach seemed in order. Only those overlays on system disk were moved to ECS. Those in the RPL were left where they were. This way the improved performance observed in the model near thrashing could still be realized and, possibly, the degradation in the balanced system might be less pronounced.

5.5.1 Experiment Description

The corresponding SPG's were modified as in the previous experiment. The one millesecond overhead was still attached to the loading of overlays from ECS to central memory.

5.5.2 Comparison of Results

Not only was the improved performance noted in SIM 1 preserved, the performance was improved even more. This indicates that the increased CPM overhead of the last experiment was not completely absorbed by the idle fraction as supposed. Further, the performance of SIM 2 was not degraded but improved slightly. The saving in PPU time can be more clearly noted in the experiment with SIM 2. It still would seem that the saved PPU time constitutes the main source of improvement for SIM 1.

MEASURE	SIM 1	EXP. E-1	SIM 2	EXP. E-2
Elapsed Time (seconds)	1541.406	1461.829	1238.074	1226.213
Utilization				
CPU				
Control Points	.3959	.4175	.6788	.6854
CPM	.2620	.2740	.2506	.2525
PPU	.9955	.9932	.8772	.8457
Central Memory	.90	.90	.91	.90
Degree of Multiprogramming	12.7	12.6	9.3	9.3
Storage Move				
Count	13763	13062	8369	8009
Rate	8.9	8.9	6.7	6.5
Job Scheduler				
Count	4629	4467	3660	3766
Rate	3.0	3.0	2.9	3.0
Punts				
Count	702	642	516	433
Rate	.455	.440	.417	.353

Experiment E - Overlays on System Disk Moved to ECS

Table 5.8

Comparison of Performance Measures

Table 5.9 presents channel statistics for the two models. Again note that the utilization of the system channel was not reduced as much as might be expected.

The non-RSY channel requests increased for SIM 1 from 10.8 to 11.8 per second. For SIM 2 the increase was somewhat less, 13.1 to 13.3 per second. Both increases are a further evidence of more useful work being accomplished.

MEASURE	SIM 1	EXP. E-1	SIM 2	EXP. E-2
RCH				
Count	13715	13909	14311	14332
Mean Waiting Time	36.02	38.42	26.22	27.19
Mean Holding Time	100.52	99.03	76.18	75.54
Utilization	.2981	.3141	.2936	.2943
Balance	.7362	.7205	.7439	.7353
RCS				
Count	3076	3350	1906	1964
Mean Waiting Time	138.22	166.00	87.06	72.67
Mean Holding Time	160.84	163.02	99.45	97.40
RSY				
Count	5652	2500	4556	1880
Mean Waiting Time	148.09	154.50	88.76	90.45
Mean Holding Time	112.35	171.18	104.56	179.80
RSY + RCH				
Count	8728	5850	6462	3844
Mean Waiting Time	144.62	161.09	88.26	81.37
Mean Holding Time	129.44	166.51	103.05	137.70
Utilization	.7329	.6664	.5379	.4317
Balance	.4723	.5083	.5386	.6286

(Time in Milleseconds)

Experiment E - All Overlays on System Disk Moved to ECS

Table 5.9

Channel Statistics Comparison

5.5.3 Summary

In the preceding two experiments, the rate of execution for 1XX was not increased to reflect the systems utilization of ECS to hold the overlays. This was because the overlays actually occupy very little storage as compared to the capacity of ECS. The slight increase in the rate of execution for 1XX would have had little effect on the performance of the model.

Experiment E would seem to indicate that such a modification to the system might be worthy of more investigation.

5.6 Experiment F - Copy of System on Each Disk

The system disk channel is the most heavily used disk channel in the system and any request for that channel results in a relatively long delay for the requesting PPU. Experiment F concerns modifying the system so that a copy of the system is on each of the four disks.

5.6.1 Experiment Description

In the model, RCH and RCS functions were combined into one, RCH. The designated channel for each RCH function encountered was determined by a random selection. The designated channel for each RSY function was a free channel or the channel with the shortest queue. To prevent any channel from being selected inordinately more often than any other, the search for a free channel or the one with the shortest queue was cyclic, each search initiating with the channel after the one selected for the previous RSY. No attempt was made to reflect the increased seek time that might be expected due to the increased disk occupancy.

5.6.2 Comparison of Results

The greatest effect of the modification was to reduce the PPU time wasted while waiting on a channel. The freed PPU time resulted in some improvement in the performance of SIM 2 and, again, a considerable improvement in SIM 1. The freed PPU time caused only a slight retreat from PPU saturation in SIM 1.

The figures in Table 5.11 indicate a great improvement in channel balance for the two models. The mean channel wait time decreased

31.8 milleseconds for SIM 1 and 18.5 milleseconds for SIM 2.

MEASURE	SIM 1	EXP. F-1	SIM 2	EXP. F-2
Elapsed Time (seconds)	1541.406	1391.992	1238.074	1213.886
Utilization				
CPU				
Control Points	.3959	.4385	.6788	.6924
CPM	.2620	.2781	.2506	.2581
PPU	.9955	.9913	.8772	.8488
Central Memory	.90	.90	.91	.92
Degree of Multiprogramming	12.7	12.5	9.3	9.4

Experiment F - Copy of System on Each Disk

Table 5.10

Comparison of Performance Measures

MEASURE	SIM 1	EXP. F-1	SIM 2	EXP. F-2
RSY				
Count	5652		4556	
Mean Holding Time	112.3485		104.5604	
Mean Waiting Time	148.9087		88.764	
RCH+ RCS				
Count	16791		16217	
Mean Holding Time	111.5752		79.051	
Mean Waiting Time	54.740		33.313	
RSY + RCH + RCS				
Count	22443	22572	20773	20709
Mean Holding Time	111.7700	111.7548	84.6461	84.8418
Mean Waiting Time	78.2521	46.4332	45.5219	27.0293
Balance	.5882	.7126	.6500	.7584

Experiment F - Copy of System on Each Disk

Table 5.11

Channel Statistics Comparison

5.6.3 Summary

The trend so far in these experiments seems well established. Modifications which reduce the PPU load cause great improvements in SIM 1. SIM 2, on the other hand, responds more to a decrease in CPM of the CPU.

Multiple copies of the system would seem to benefit both a clogged and balanced system. The results obtained in this experiment would be moderated, of course, due to the increased disk loading.

5.7 Experiment G - Centralized CIO

The saturation of PPU's constitutes the major bottleneck in system performance for SIM 1. Three approaches to minimize the restrictions imposed by PPU saturation are:

(1) use the PPU's more efficiently;

(2) shift some of the processing load from the peripheral processors to the central processor; and

(3) increase the number of PPU's.

The experiments so far have tended to utilize approach (1). This experiment will utilize both (1) and (2).

CIO is the most often executed peripheral process in the UT-2 system. Its basic duties are to determine the validity of the request, reserve the local file and select one of its overlays to process the request. Approximately 60% of all CIO executions result in data being transferred to or from disk or tape. When a data transfer is to occur, CIO and its overlays occupy a PPU on the order of 15 milleseconds prior to issuing the request for a channel. During Run 1, a PPU issuing a request for a non-system disk channel had to wait an average of 36.4947 milleseconds before being allocated the channel and, if the request were for the system channel, the mean waiting time was 135.8679 milleseconds. The corresponding times observed for Run 2 were 23.8348 and 83.1569 milleseconds (see Tables 4.3 and 4.4).

Experiment G is an attempt to reduce the amount of PPU time wasted waiting on channels. Processing done by CIO and that done by its overlays from entry to the point that the first request channel function was encountered was moved to CPM. The remaining portions of the I/O processes were still executed on PPU's. An I/O process could not, however, enter the queue for a PPU until its desired channel was free.

5.7.1 Experiment Description

In the model, when a CIO request was encountered, the SPG's for CIO and its overlays were traversed until a release PPU or a request channel function was issued, keeping track of the processing time requirements. To resolve the differences in execution speed between the CPU and PPU's, the time required for CPM to perform tasks done by PPU's was approximated by dividing the PPU processing time requirement by 10. The proper events in the event list would be perturbed by this amount of time to reflect the execution of CPM. If a request channel function were encountered, the channel would be determined and the I/O process entered into a FCFS queue for that channel. When the channel is, or becomes, free, the next process in the queue enters the PPU queue ahead of all other processes except 1SJ and 1RJ. Only one CIO type I/O process can be in the PPU queue at a time for any given channel, the other would remain in the queue for that channel. Once a PPU is assigned to the process, the remaining portion of the SPG for that process is traversed in the normal manner, beginning with the issuance of the request channel function.

5.7.2 Comparison of Results

Due to PPU saturation in SIM 1, a large improvement in system performance was expected and realized. The almost 2% decrease in PPU utilization was reflected in the over 6% increase in CPU utilization by control points as illustrated in Table 5.12. The experiment also indicated an increase in CPU utilization by control points in SIM 2. As the PPU's were not saturated, a large portion of the improvement in SIM 2 can probably be attributed to the decrease in CPU overhead attributed to CPM.

MEASURE	SIM 1	EXP. G-1	SIM 2	EXP. G-2
Elapsed Time (seconds)	1541.406	1317.850	1238.074	1176.431
Utilization				
CPU				
Control Points	.3959	.4631	.6788	.7144
CPM	.2620	.2693	.2506	.2268
PPU	.9955	.9775	.8772	.8169
Central Memory	.90	.89	.91	.90
Degree of Multiprogramming	12.7	13.2	9.3	9.4

Experiment G - Centralized CIO

Table 5.12

Comparison of Performance Measures

A close study of Table 5.13 indicates that more useful work got accomplished by each job per residency period in central memory. The rate of execution of storage moves and the job scheduler decreased markedly, contributing to the low CPM overhead mentioned above. The rate of jobs being "rolled in" and "rolled out" also decreased slightly. Table 5.14 presents a more revealing comparison of the system overhead features as the rates are adjusted to reflect the number of executions per second of CPU time used by control points (useful work).

The CPM overhead per CIO for SIM 1 was 3.6 milleseconds, for SIM 2 it was 2.84 milleseconds. The actual increase in overhead for

MEASURE	SIM 1	EXP. G-1	SIM 2	EXP. G-2
Executions Per Second				
Job Scheduler	3.0	2.2	2.9	2.1
Storage Move	8.9	8.25	6.7	6.3
1SJ	2.33	2.21	2.01	1.88
1RJ	2.79	2.64	2.42	2.22
Number of Punts Per Second	.455	.426	.417	.343

Experiment G - Centralized CIO

Table 5.13

Job Scheduler and Storage Move Rates

MEASURE	SIM 1	EXP. G-1	SIM 2	EXP. G-2
Executions Per Second*				
Job Scheduler	7.58	4.71	4.35	2.94
Storage Move	22.55	17.80	9.96	8.82
1SJ	5.88	4.78	2.96	2.64
1RJ	7.04	5.70	3.57	3.11
Number of Punts Per Second*	1.150	.920	.614	.480

*Second of CPU time used by control points

Experiment G - Centralized CIO

Table 5.14

Job Scheduler and Storage Move Rates (adjusted)

CIO's is less than these figures as the CPM always had to process various functions for CIO and its overlays.

The increased utilization of the CPU by users also resulted in higher I/O rates as illustrated in Table 5.15.

MEASURE	SIM 1	EXP. G-1	SIM 2	EXP. G-2
Utilization				
System Disk Channel	.7329	.8425	.5378	.5578
Non-System Disk Channel	.2981	.3126	.2936	.3006
Disk Channel Requests per second	14.56	17.52	16.70	17.09

Experiment G - Centralized CIO

Table 5.15

Disk Channel Statistics

5.7.3 Summary

This particular experiment displays the capacity provided by the model to investigate rather large modifications to the system software and isolate those areas which reflect those changes. This experiment required a considerable amount of software modification in the model and some revalidation. The revalidation consisted primarily of standard debugging and insuring that the modified model reflected the desired change.

The large improvement in system performance supports this modification as a candidate for actual implementation.

5.8 Experiment H - 13 PPU's

In this experiment the third approach mentioned in the previous section was utilized in an attempt to alleviate the PPU saturation problem. Experiment H consists of investigating the behavior of the model were the number of PPU's increased from 10 to 13.

5.8.1 Comparison of Results

The results of this experiment are quite similar to those in the preceding section. A considerable improvement was again seen in SIM 1, as expected. Improvement was also seen in SIM 2. To demonstrate the similarities in experiments G and H, performance utilization statistics for both are listed in Tables 5.16 and 5.17. As part of the PPU processing in experiment G was moved to central memory, comparing the PPU utilization figures for the two experiments reveals little. Note the relation between the utilization of the CPU by control points and by CPM for SIM 2 in Table 5.17. This supports the implication that the CPU is the primary bottleneck for SIM 2.

The similarity between the two experiments extends to the I/O rate as can be seen in Table 5.18.

A comparison of Tables 5.13 and 5.19 indicates that the amount of useful work per period of memory residency was also quite close.

MEASURE	SIM 1	EXP. H-1	EXP. G-1
Elapsed Time	1541.406	1287.016	1317.850
Utilization			
CPU			
Control Points	.3959	.4742	.4631
CPM	.2620	.2701	.2693
PPU	.9955	.9476	.9775
Central Memory	.90	.86	.89
Degree of Multiprogramming	12.7	12.01	13.2
Mean Number of PPU's Occupied	9.955	12.319	9.775

Experiment H - 13 PPU's

Table 5.16

Comparison of Basic Processor Utilization

MEASURE	SIM 2	EXP. H-2	EXP. G-2
Elapsed Time	1238.074	1187.242	1176.431
Utilization			
CPU			
Control Points	.6788	.7079	.7144
CPM	.2506	.2498	.2268
PPU	.8772	.7127	.8169
Central Memory	.91	.90	.90
Degree of Multiprogramming	9.3	9.2	9.4
Mean Number of PPU's Occupied	8.772	9.265	8.169

Experiment H - 13 PPU's

Table 5.17

Comparison of Basic Processor Utilization

MEASURE	EXP. G-1	EXP. H-1	EXP. G-2	EXP. H-2
Utilization				
System Disk Channel	.8425	.8533	.5378	.5494
Non-System Disk Channel	.3126	.3513	.2936	.3016
Disk Channel Requests per second	17.52	17.59	17.09	16.82

Experiment H - 13 PPU's .

Table 5.18

Disk Channel Statistics

MEASURE	SIM 1	EXP. H-1	SIM 2	EXP. H-2
Executions Per Second				
Job Scheduler	3.0	2.24	2.9	2.78
Storage Move	8.9	8.9	6.7	6.6
1SJ	2.33	2.32	2.01	2.01
1RJ	2.79	2.77	2.42	2.41
Number of Punts Per Second	.455	.453	.417	.401

Experiment H - 13 PPU's

Table 5.19

Job Scheduler and Storage Move Rates

5.8.2 Summary

These last two experiments had extremely similar effects on performance of the model, providing an option if the problem of PPU saturation were to be actually attacked.

Experiment H demonstrates the utility of the model in investigating hardware, as well as software, modifications.

CHAPTER 6

SUMMARY

The primary goal of this research was to develop a trace-driven model of an actual operating system capable of primitive process level simulation. This goal was attained by representing the elementary processes of the system by directed graphs which explicitly reflected their resource utilization patterns and provided a tractable method for handling their interactions and interrelationships. The set of graphs required for this study was obtained through the analysis of data recorded by the UT-2 trace monitor. Data from the trace was also used to parameterize and validate the model.

The utility of the model was then demonstrated by several experimental studies reflecting various levels of system modification. Some modifications were localized functional changes and, while no real perturbations were reflected in the gross system performance measures, localized consequences were isolated in other areas of the system. Even with larger scale modifications which had more effect on overall system performance, the model still allowed localized changes to be isolated, often providing more insight as to why a modification had some particular effect.

As the model was explicitly graph driven, the functional structure was not only easily altered, but also required little or no re-validation when altered. This represents no small gain as the re-validation of a modified model during experimentation is often an arduous task.

While the experimental studies were primarily to demonstrate the versatility and flexibility of the model, two modifications do seem particularly worthy of further study. The first would be that of moving the code for peripheral processes which resides on the system disk to ECS. The centralized CIO modification would be the other.

Due to the resolution and flexibility provided, the developing and maintaining of such a graph structured model as a working tool for system analysis and refinement would be of practical utility to a system design and analysis shop.

BIBLIOGRAPHY

[1] Brice, R., A Study of Feedback Coupled Resource Allocation Policies in a Multiprocessing Computer Environment, Computation Center and Department of Computer Sciences TSN-35, (Ph.D. Dissertation), The University of Texas at Austin, August 1973.

[2] Control Data Corporation. Control Data 64/65/6600 Computer System Reference Manual, Publication Number 60100000, 1967.

[3] Control Data Corporation. Chippewa Operating System Reference Manual, Publication Number 60134400, 1966.

[4] Dijkstra, E.W., "The Structure of 'The'-Multiprogramming System," Communications of the ACM 11, Number 5 (May 1968), 341-346.

[5] Dissly, C.W., The Design and Implementation of System I/O on the CDC 6600 at the University of Texas at Austin, Computation Center TSN-22, The University of Texas at Austin, July 1971.

[6] Hebalkar, P.G., Deadlock-Free Sharing of Resources in Asynchronous Systems, Project MAC TR-75, (D.Sc. Thesis), Massachusetts Institute of Technology, September 1970.

[7] Holt, R.C., On Deadlock in Computer Systems, Technical Report CSRG-6, (Ph.D. Dissertation), Computer Systems Research Group, University of Toronto, April 1971.

[8] Holt, R.C., "Comments on Prevention of System Deadlocks," Communications of the ACM 14, Number 1 (January 1971).

[9] Howard, J.H., The Coordination of Multiple Processes in Computer Operating Systems, Computation Center TSN-16, (Ph.D. Dissertation), The University of Texas at Austin, December 1970.

[10] Howard, J.H., "A Large-Scale Dual Operating System," Proceedings ACM National Conference, 1973, 242-248.

[11] Howard, J.H. and W.P. Alexander, "Analyzing Sequences of Operations Performed by Programs," Program Test Methods, Prentice Hall, 1973, 239-254.

[12] Johnson, D.S., A Process-Oriented Model of Resource Demands in Large, Multiprocessing Computer Utilities, Computation Center and Department of Computer Sciences TSN-29, (Ph.D. Dissertation), The University of Texas at Austin, 1972.

[13] Noe, J.D. and G.J. Nutt, "Validation of a Trace-Driven CDC 6400 Simulation," Proceedings Spring Joint Computer Conference (1972), 749-757.

[14] Noetzel, A.S., "The Design of a Meta-System," Proceedings Spring Joint Computer Conference (1971), 415-424.

[15] Noetzel, A.S., The Design of a Meta-System for Measurement and Simulation of Timesharing Computers, Report 71-15, University of Pennsylvania, February 1971.

[16] Rodriguez, J.E., A Graph Model for Parallel Computations, Project MAC TR-56, Massachusetts Institute of Technology, September 1969.

[17] Sherman, S.W., Trace-Driven Modeling Studies of the Performance of Computer Systems, Computation Center and Department of Computer Sciences TSN-30, (Ph.D. Dissertation), The University of Texas at Austin, August 1972.

[18] Sherman, S.W., J.C. Browne and F. Baskett, "Trace Driven Modeling and Analysis of CPU Scheduling in a Multiprogramming System," Proceedings of ACM Workshop on Performance Evaluation, Cambridge, Massachusetts, April 1971, 173-199.

[19] Sherman, S.W., J.H. Howard and J.C. Browne, "A Comparison of Deadlock Prevention Schemes Using a Trace Driven Model," Proceedings of 6th Princeton Conference on Information Sciences and Systems, Princeton, New Jersey, March 1972.

[20] Schwetman, H.D., A Study of Resource Utilization and Performance Evaluation of Large-scale Computer Systems, Computation Center TSN-12, (Ph.D. Dissertation), The University of Texas at Austin, August 1970.

[21] Thornton, J.E., Design of a Computer: The CDC 6600, Scott, Foresman and Co., Glenview, Illinois, 1970.

[22] Wedel, W.M., An Introduction to UT-2D for System Programmers, Computation Center TIMS-7, The University of Texas at Austin, 1973.