AD-A100 130 MARYLAND UNIV COLLEGE PARK DEPT OF COMPUTER SCIENCE F/G 9/2
MODELS OF SWAPPING.(U)
JAN 81 L W DOWDY, H J BREITENLOHNER AFOSR-78-3654
UNCLASSIFIED TR-1004 AFOSR-TR-81-0482 NL

1 of 1
AD
A100130

END
DATE
FILMED
7 81
DTIC

LEVEL II (12)

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-81-0482

2. GOVT ACCESSION NO.: AD-A100130

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): MODELS OF SWAPPING.

5. TYPE OF REPORT & PERIOD COVERED: INTERIM

6. PERFORMING ORG. REPORT NUMBER: TR-1004

7. AUTHOR(s): Lawrence W. Dowdy ~~and~~ Hans J. Breitenlohner

8. CONTRACT OR GRANT NUMBER(s): AFOSR-78-3654

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Computer Science
University of Maryland
College Park MD 20742

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
PE61102F
2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS:
Air Force Office of Scientific Research/NM
Bolling AFB DC 20332

12. REPORT DATE: JANUARY 1981

13. NUMBER OF PAGES: 18

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Swapping; closed queueing networks; parameter interdependencies; performance prediction; model validation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The performance of a computer system depends upon the efficiency of its swapping mechanisms. The swapping efficiency is a complex function of many variables. The degree of multiprogramming, the relative loading on the swapping devices, and the speed of the swapping devices are all interdependent variables that affect swapping performance.
(CONT.)

ITEM #20, CONT.:

In this paper, a model of swapping behavior is given. The interdependence between the degree of multiprogramming, the swapping devices loadings, and the swapping devices speeds are modeled using an iterative scheme.

The validation of a model is its predictive capability. The given swapping model was applied to an actual system to predict the effect of moving the swapping activity from drums to discs. When the swapping activity was actually moved, throughput increased by 20%. The model accurately predicted this improvement.

Subtopics discussed include: (1) the modeling of blocked and overlapped disc seek activity; (2) the usefulness of empirical formulae; and (3) the calibration of unmeasurable parameters. Extensions and further applications of the model are given.

Technical Report TR-1004
AFOSR-78-3654
CSC

January 1981

Models of Swapping

by

Lawrence W. Dowdy
Hans J. Breitenlohner

Models of Swapping

Lawrence W. Dowdy and Hans J. Breitenlohner
University of Maryland
College Park, Maryland 20742

Abstract

The performance of a computer system depends upon the efficiency of its swapping mechanisms. The swapping efficiency is a complex function of many variables. The degree of multiprogramming, the relative loading on the swapping devices, and the speed of the swapping devices are all interdependent variables that affect swapping performance.

In this paper, a model of swapping behavior is given. The interdependencies between the degree of multiprogramming, the swapping devices' loadings, and the swapping devices' speeds are modeled using an iterative scheme.

The validation of a model is its predictive capability. The given swapping model was applied to an actual system to predict the effect of moving the swapping activity from drums to discs. When the swapping activity was actually moved, throughput increased by 20%. The model accurately predicted this improvement.

Subtopics discussed include: 1) the modeling of blocked and overlapped disc seek activity, 2) the usefulness of empirical formulae, and 3) the calibration of unmeasurable parameters. Extensions and further applications of the model are given.

Key Words and Phrases: swapping, closed queuing networks, parameter interdependencies, performance prediction, model validation.

This research was supported in part by the Air Force Office of Scientific Research under Grant AFOSR-78-3654 and also in part by the Computer Center of the University of Maryland.

Outline

1. Introduction
2. Modeling Approach
   - 2.1 Model Construction
   - 2.2 Model Parameterization
   - 2.3 Overview
3. Validation <===> Prediction of Parametric Intercependencies
   - 3.1 Parameter Prediction Scheme
   - 3.2 Results
4. Conclusions

1. Introduction

Queuing network models of computer systems have increased in popularity within the past few years [CS 78] [Comp 80]. These models are being constructed for an increasing range of problems (e.g., memory modeling [BBC 77], job class modeling [KT 78], passive resource modeling [SM 79], software modeling [SB 80], overhead modeling [KKT 80]). Exact and approximate solution techniques of these models have multiplied [BCMP 75] [DB 78] [RL 80] [Trip 79] [CS 80] [Zaho 80] [CN 80]. However, the motivations for constructing and solving these queuing network models are practical applications: tuning actual systems and predicting performance if some system parameters change.

Predicting system performance when the system parameters are changed is difficult. Constructing a model of an observation period is not the problem. Neither is the solution of the model the difficulty. The problem is exposing and modeling the effects upon all system parameters when a particular system parameter is altered. For example, if a system is upgraded by adding another CPU processor, the degree of multiprogramming, the amount of swapping activity, and the effective I/O device speeds are all likely to change [DACT 79]. The interdependencies among the system parameters must be understood and modeled.

In this paper, the parameter interdependencies relating to the swapping activity within a computer system are explored. A modeling approach for constructing models of swapping is given in Section 2. The prediction of parameter interdependencies under a system change is presented and validated in Section 3. A summary and further applications of the swapping models are discussed in Section 4.

2. Modeling Approach

The presentation of the modeling approach is facilitated by describing an explicit example. In Section 2.1, the queuing network model example is constructed. Parameterization of this model is done in Section 2.2. Since some ot the parameters are unavailable from measurement, an iteration/calibration technique is described. The parameterization of the discs leads to a submodel of the blocking and overlapping disc seek activity. In Section 2.3, after the construction and parameterization of the example, an overview of the modeling approach is given.

2.1 Model Construction

The example system studied is a Univac 1100/42 running under EXEC8 level 36 operating system. A central server queuing model describing the system is shown in Figure 1.

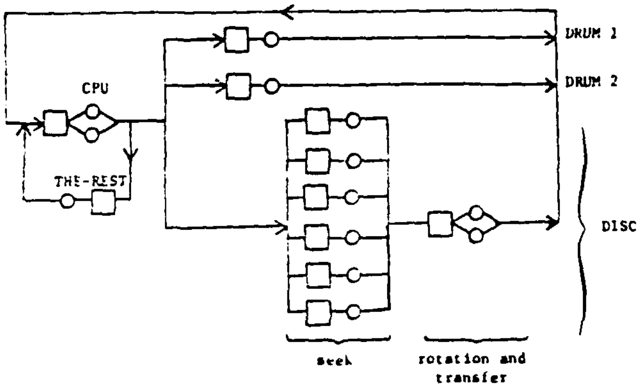

Figure 1: Swapping Model

Standard queuing theoretical assumptions are made [CS 78]. The CPU queue is processor shared while all other queues are serviced first-come-first-served. The CPU server, the DISC rotation (i.e., latency) and transfer server, and the DISC seek servers are load dependent servers. The CPU server is load dependent since the queue is serviced by two processors. The DISC rotation and transfer server is load dependent since the queue is serviced by two channels. The DISC seek servers, one per disc unit, are load dependent since a particular disc cannot be seeking and transferring at the same time. As the load increases at a disc, the seek server is blocked more frequently, thus delaying pending requests. The drum servers and THE-REST server are load independent servers. A fixed number of customers (i.e., degree of multiprogramming (DMP), jobs, transactions) flow through the network based upon given branching probabilities (i.e., visit ratios) of requiring service at the various servers. Calculation of the parameters (i.e., DMP, branching probabilities, service rates, load dependency factors) is described in Section 2.2.

The motivation for constructing the model as shown in Figure 1 is to accurately model the swapping activity. Feasible swapping devices include only the drums and the discs. These devices are modeled in detail. The remaining peripheral devices (e.g., tapes, printers, readers, terminals) are not the subject of analysis and, therefore, their effects on swapping are grouped together in one composite server, THE-REST. [Note: THE-REST server is not a composite server in the decomposition theory sense [CHW 75], since load dependent service rates are unavailable for THE-REST. The load independent service rate for THE-REST is found using an iteration/calibration technique described in Section 2.2.]

2.2 Model Parameterization

The model derives its parametric values from the Univac Software Instrumentation Package. This software monitor provides the following information for each channel:

- active time
- number of transactions by file type (e.g., user files, swapping activity files, directory lookup files, etc.)
- mean wait time per transaction
- mean number of words transferred per transaction by file type.

The parameterization for each server in Figure 1 follows.

CPU

The CPU parameters consist of the load dependent service rates. The CPU server is load dependent since two processors service the CPU queue.

$$\text{CPU service rate (one in queue)} = \frac{\text{the number of transactions processed by the two CPU processors}}{\text{CPU processor 1 active time + CPU processor 2 active time}}$$

$$\text{CPU service rate (two or more in queue)} = 2 * \text{CPU service rate (one in queue)}$$

DRUMs

For each drum, the parameters consist of the branching probability to the device and the load independent mean service rate.

$$\text{DRUM branching probability} = \frac{\text{the number of transactions processed by the drum}}{\text{total number of transactions processed by the CPU}}$$

$$\text{DRUM service rate} = \frac{\text{the number of transactions processed by the drum}}{\text{active time of the drum}}$$

The drum service time is the sum of two components, access time plus transfer time. The mean transfer time is computed as the ratio of the mean number of words transferred per transaction to the channel speed (240 words/ms from hardware specifications). Since the mean drum access time is an unmeasurable quantity, it is computed as:

$$\begin{aligned}\text{DRUM access time} &= \text{DRUM service time} - \text{DRUM transfer time}\\ &= \frac{1}{\text{DRUM service rate}} - \frac{\text{the number of words per transaction to the drum}}{\text{drum channel speed}}\end{aligned}$$

The drum access time is necessary for prediction purposes. As swapping activity changes, the mean number of words per transactin to the drum changes. This changes the drum's mean service time. However, the drum's access time and the channel speed are assumed to be invariant. That is, once the drum access time is found, the predicted drum service rate is a function of only the mean number of words transferred per transaction to the drum.

DISC Rotation and Transfer

The parameters of the DISC rotation/transfer server consist of the load dependent service rates. The server is load dependent since two channels service the queue. Two items are assumed invariant: 1) the mean rotation time per transaction which is roughly one half a disc revolution (8.3 ms from hardware specifications) and 2) the channel speed (179 words/ms from hardware specifications).

$$\text{DISC rotation and transfer service rate (one in queue)}$$

$$= \frac{1}{\text{DISC rotation time} + \text{DISC transfer time}}$$

$$= \frac{1}{\text{DISC rotation time} + \frac{\text{the number of words per transaction to the disc}}{\text{disc channel speed}}}$$

$$\text{DISC rotation and transfer service rate (two or more in queue)}$$

$$= 2*\text{DISC rotation and transfer service rate (one in queue)}$$

DISC Seek

The parameters of each DISC seek server are the branching probability to the device and the load dependent service rate.

$$\text{DISC seek branching probability} = \frac{\text{the number of transactions processed by the disc}}{\text{total number of transactions processed by the CPU}}$$

The disc seek load dependent service rates are not straightforward. The servers cannot be assumed to be load independent. A particular disc request causes a device seek followed by rotation and channel transfer. (For the moment, other components such as the channel time to initiate the seek are ignored.) The seek, rotation, and transfer activities engage the disc device but only the rotation and transfer activities engage the channel. While a particular disc device is seeking, parallel seeking or transferring on different disc devices is possible. However, while seeking on a particular device, no rotation or transfer from the device is possible, and vice versa. Therefore, if two requests are queued at a particular disc device, the first request is serviced and can then proceed to the rotation and transfer queue, while the second seek request must wait until the first request has finished transferring. While rotating and transferring from a particular disc device, the disc is blocked from seeking.

Queuing network models which allow blocking do not have product form solutions. The approach used here is to replace the blocking by load dependent servers. When only one request is at a particular disc, it can be serviced by the seek server at a rate dependent upon the movement speed of the read/write arm (30 ms from hardware specifications). When two or more requests are at a particular disc, the first request can proceed normally since it experiences no extra delays (i.e., blocking), but the pending requests are delayed. The amount of extra delay is the expected waiting time experienced at the rotation and transfer queue.

$$\text{DISC seek service rate (one in queue)} = \frac{1}{\text{time to move read/write arm}}$$

$$\text{DISC seek service rate (two or more in queue)}$$
$$= \frac{1}{\text{time to move read/write arm} + \text{average waiting time at the rotation and transfer server}}$$

The average waiting time at the rotation and transfer server is a function of the load placed upon the discs. To get an approximation for this waiting time, we assume the rotation and transfer server behaves as an M/M/2 queue. Therefore,

$$\text{average waiting time at the rotation and transfer server} = \frac{4\mu}{4\mu^2 - \lambda^2}$$

$$\text{where } \mu = \text{DISC rotation and transfer service rate (one in queue)}$$

$$\text{and } \lambda = \frac{\text{the number of transactions processed by all the discs}}{\text{observation time period}}$$

Two extensions can be easily made which make the disc seek service rates more accurate. First, when only one customer is in queue, the time needed to acquire the channel initially to initiate the seek can be included. If the channels are both busy, this time is the mean time until a channel finishes transferring its current request. If one channel is idle, the time required to obtain a channel to initiate a seek is zero. Second, when two or more customers are in queue, the extra delay spent as a result of blocking is, 1) the expected waiting time in only the queue at the rotation and transfer server, plus 2) the expected transfer time of a request to the particular blocked disc. This utilizes the information that different discs have different mean file sizes (i.e., different average number of words transferred per transaction) and thus experience different transfer times. That is, a large file transfer blocks a disc longer than a small file transfer.

Two items should be noted: 1) the model of load dependent disc seek servers is an approximation of the actual blocking situation, and 2) in the model when two or more transactions are at the seek server, the first transaction is delayed an extra amount of time and the final transaction is not, instead of the other way around as in practice. Other approximations for this channel/device blocking are available [ZHS 78] [KKT 80] [Bard 80] [Hunt 81].

THE-REST and the DMP

The parameters of THE-REST server consist of the branching probability to the server and its load independent mean service rate.

$$\text{THE-REST branching probability} = \frac{\text{the number of transaction processed by the CPU that are not processed by the drums or discs}}{\text{total number of transactions processed by the CPU}}$$

The mean service rate of THE-REST is unmeasurable since it is a composite rate of many devices which are not explicitly modeled. The degree of multiprogramming, DMP, is not accurately measured by the software monitor. These two parameters are found by using the following iteration/calibration technique.

1. Guess an initial value for the DMP.

2. Guess an initial value for the mean service rate of THE-REST.

3. Solve the queuing network model for the throughputs of each server.

4. If the modeled throughputs do not match the observed throughputs on the explicitly modeled devices, change the mean service rate of THE-REST. If the model overestimates (underestimates) the throughputs, decrease (increase) the mean service rate of THE-REST.

5. Iterate on steps 3 and 4 until the modeled and observed throughputs match.

6. Check the modeled mean wait times at the explicitly modeled devices. If they do not match the observed mean wait times, change the DMP. If the model overestimates (underestimates) the mean wait times, decrease (increase) the DMP.

7. Iterate on steps 2 through 6 until the modeled and observed throughputs and wait times match.

As an example of the parameterization, a benchmark was constructed with all swapping allocated to the drums, which is what is done in practice. Figure 2 gives the parameterized model.

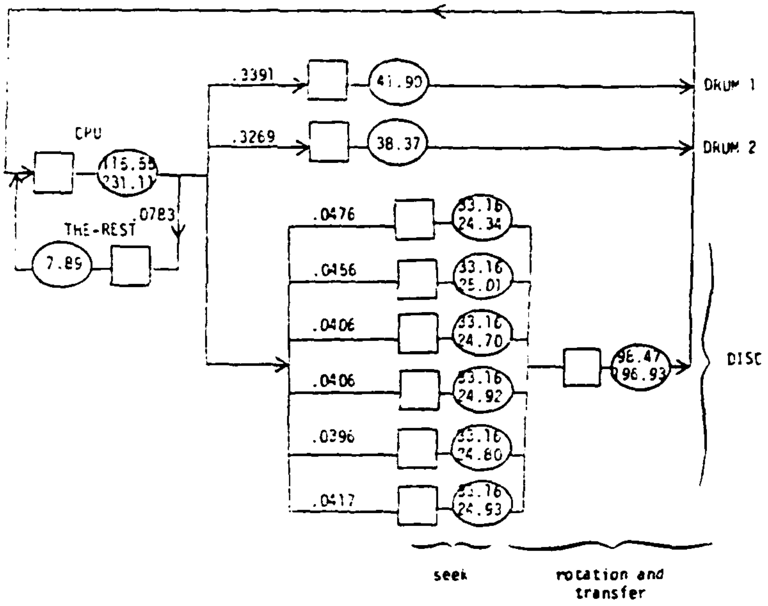

Figure 2: Parameterized Model -- Swapping to the Drums

Modeled versus observed performance is shown in Table 1 and demonstrates the accuracy of the parameterization techniques described.

Table 1: Modeled and Observed Performance

parameter	modeled	observed	% error
DRUM 1 throughtput (transactions per second)	34.3	34.3	0.0
DRUM 1 utilization (percent)	81.8	81.7	0.1
DRUM 1 wait time (milliseconds)	124.0	126.2	1.7
DRUM 2 throughput	33.0	33.0	0.0
DRUM 2 utilization	86.1	86.1	0.0
DRUM 2 wait time	169.3	167.3	1.2
DISC throughout	25.8	25.8	0.0
CPU throughput	101.0	101.0	0.0

2.3 Overview

The primary theme in the modeling approach is to explicitly and accurately model the possible swapping devices. The effects that other peripheral devices have upon swapping are modeled via a composite server whose parameters are found using iteration/calibration. An overview procedure of the modeling approach presented is:

1. Construct a queuing network model of the system. Identify those servers that may be used for swapping activity. Model the swapping servers' activities in detail.

2. Determine the service rate characteristics (possibly load dependent) of the CPU server.

3. For each server that may be used for swapping activity:

   a) Determine the relative loading (i.e., visit ratios, branching probabilities).

   b) Determine the service rate characteristics (possibly load dependent). The service rates should be a function of the average number of words transferred per transaction. This allows a mechanism to easily model swapping activity changes since a change in the swapping activity alters the average number of words per transaction as seen by the device.

4. Group the remaining non-swapping devices into a single server. Determine the relative loading to this composite server.

5. Use the unmeasurable parameters (e.g., the service rate characteristics of the composite server, the degree of multiprogramming) as calibration variables to, as closely as possible, match observed performance measures to modeled performance measures.

3. Validation <===> Prediction of Parametric Interdependencies

The validation of any modeling approach is its predictive capability. It is one issue to construct a model which faithfully reproduces performance measures which are based upon observed performance measures. It is quite another issue to independently alter the model and the actual system and still have consistent performance between the two. This second issue addresses model validation.

Irrespective of validation, the predictive capability of a model has a second purpose -- application. Primary uses of a model include answering "what if" questions and illustrating system inefficiencies.

This section addresses both the validation and the application of the modeling approach presented in the previous section. The benchmark example of the previous section indicates overutilization of the drums and underutilization of the discs. One component of the drum activity is the swapping activity. The "what if" question is: what would happen to system performance if all the swapping activity were moved to the discs? In practice, the swapping activity is typically assigned to the "fastest" devices (e.g., the drums), while the "slower" devices (e.g., the discs) are underutilized. However, due to multiple channeling and parallelism, these "slower" devices can provide comparable or superior performance. In order to answer the "what if" question accurately, the parametric interdependencies when the system is altered are identified and modeled. The benchmark is rerun on the altered system. A degree of model validation is attained once predicted and observed performance agree. Section 3.1 outlines a procedure to predict parameter interdependencies. Section 3.2 presents results from the procedure's application.

3.1 Parameter Prediction Scheme

The system change under consideration here is moving all swapping activity from the drums to the discs. At first glance, moving the swapping activity from drum to disc appears unwise since drums are "faster" devices. However, considering overlapped disc seek times and comparable rotation and transfer times, swapping to disc becomes a viable option.

An iterative scheme is given to predict the interdependencies among the parameters when swapping activity is moved from the drums to the discs. From Section 2, a parameterized base model of the system when the swapping activity is assigned to the drums is assumed (i.e., Figure 2).

1. Assume that the DMP, CPU service rate, and THE-REST service rate remain invariant. Recall that the DMP and THE-REST service rate are unmeasured parameters and are found via iteration/calibration in the base model.

2. Assume, for the moment, that the amount of swap activity (i.e., the number of swap transactions) remains the same as in the base model.

3. Predict new branching probabilities. This is not a trivial step. The amount and location of available disc space is found and the proportion of the swap activity moved from the drums to the individual discs is predicted based upon the amount and location of this available disc space. Since the swapping activity is moved from the drums, the drums have more free space. Most systems have a "pecking order" to search among the devices for dynamic (e.g., temporary) file space. Drums are typically high in this pecking order. Thus, when the swapping activity is moved from the drums to the discs, many dynamic files automatically migrate to the drums to utilize its free space. These dynamic file characteristics must be identified since they influence the new branching probabilities.

4. Predict the new average number of words per transaction to each device. Step 3 is essentially predicting the new file assignment. Knowing which files are on which devices and knowing the file access frequencies, the new average number of words per transaction to each device can be calculated. Swapping activities invariably have a large number of words associated with each swap transaction, while user and dynamic file activities usually have a smaller number of words transferred per transaction.

5. Calculate new service rates for the drum and the rotation and transfer disc servers. These rates are strictly a function of the average number of words transferred per transaction found in step 4. Recall that the drum access time, the drum channel speed, the disc rotation time, and the disc channel speed are all assumed invariant (see Section 2.2).

6. Guess a throughput value for the discs (i.e., λ in Section 2.2).

7. Calculate the average wait time at the rotation and transfer disc server. Knowing the throughput and the service rate at the server, the M/M/2 analysis of Section 2.2 is applicable.

8. Calculate the service rates for the disc seek servers. Recall that these rates are load dependent based upon the average wait time at the rotation and transfer disc server.

9. Solve the model. All parameters have now been predicted and predicted performance measures can be obtained (e.g., [BB 80]).

10. If the predicted throughput for the discs does not match the guessed value in step 6, iterate on steps 6 through 9 using the new predicted throughput.

11. If the average number of customers in the model, excluding THE-REST server, decreases (increases), then swapping activity will likewise decrease (increase). If the DMP in the main central server model decreases, fewer customers compete for memory and the swapping activity decreases [DAGT 79]. Predict the amount of decrease (increase) from the empirical formula:

1% DMP decrease(increase) $\simeq$ 1.5% swapping rate decrease(increase)

This formula is derived from the following graph. A series of over 100 observations of the actual machine on typical days were made to observe the relationship between the DMP and the amount of swapping activity.

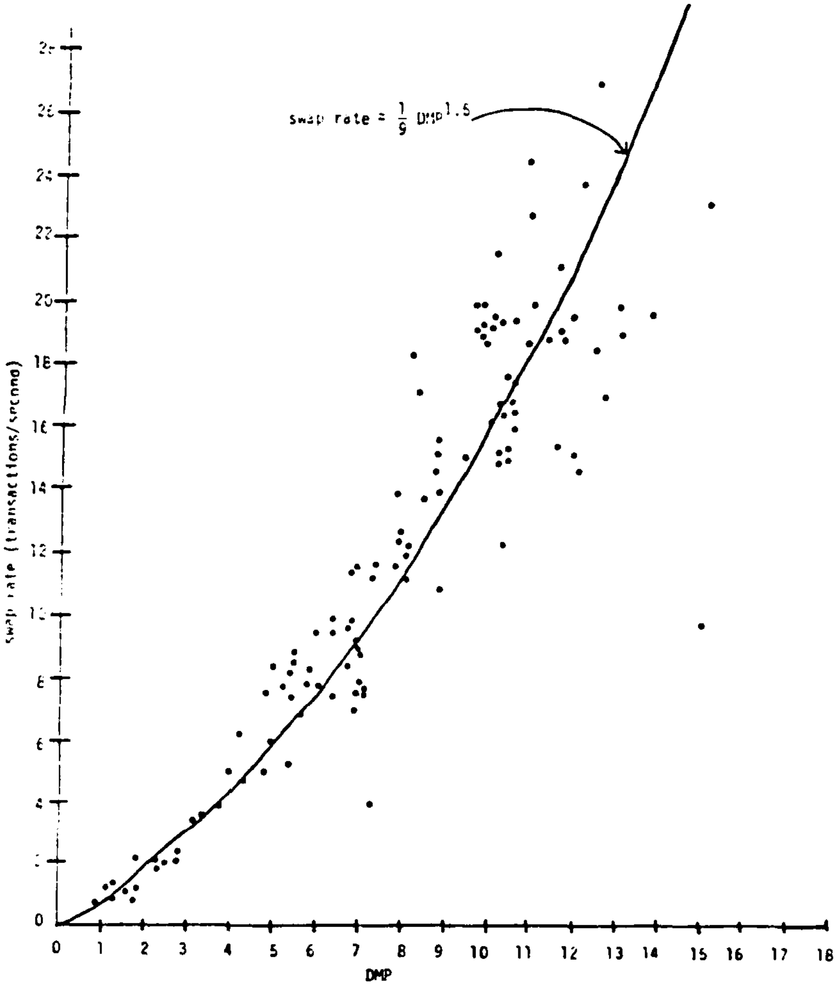

Figure 3: DMP/Swapping Relationship

12. Iterate on steps 3 through 12 changing branching probabilities, average numbers of words transferred per transaction, and service rates for each device. Terminate when the DMP in the main central server model (excluding THE-REST server) does not significantly change.

3.2 Results

The iterative scheme presented in the previous section was applied to the base parameterized model when swapping activity was assigned to the drums (see Figure 2). The swapping activity was assigned to the discs and the benchmark was rerun. Table 2 gives the results.

Table 2: Predicted Versus Actual Comparisons

parameter	swap to drum	swap to disk	% change	prediction swap to disk	% prediction error
CPU μ	115.55,231.11	117.99,235.97	2.1	115.55,231.11	2.1
TPUT	101.01	121.21	20.0	115.02	5.1
UTIL	.6085	.6790	11.6	.6649	2.1
QLEN	1.08	1.39	28.7	1.32	5.0
WAIT	10.7	11.5	7.5	11.5	0.0
DRUM 1 μ	41.90	48.66	16.1	48.93	0.6
p	.3391	.2296	32.3	.2148	6.4
TPUT	34.25	27.83	18.7	24.71	11.2
UTIL	.8175	.5719	30.0	.5050	11.7
QLEN	4.25	1.33	68.7	1.02	23.3
WAIT	124.0	47.7	61.5	41.1	13.8
DRUM 2 μ	38.37	49.31	28.5	50.93	3.3
p	.3269	.3698	13.1	.3968	7.3
TPUT	33.02	44.80	35.7	45.64	1.9
UTIL	.8606	.9090	5.6	.8962	1.4
QLEN	5.59	7.73	38.3	7.10	8.2
WAIT	169.3	172.3	1.8	155.6	9.7
DISK p	.2557	.3316	29.7	.3199	3.5
TPUT	25.83	40.19	55.6	36.80	8.4
QLEN	1.25	4.07	255.6	4.03	1.0
WAIT	48.5	101.2	108.7	109.4	8.1
THE-REST μ	7.98	8.51	6.6	7.98	6.2
p	.0783	.0690	11.9	.0685	0.7
TPUT	7.91	8.36	5.7	7.88	5.7
UTIL	.9912	.9828	0.8	.9874	0.5
QLEN	16.83	14.49	13.9	15.54	7.2
WAIT	2127.5	1732.2	18.6	1971.4	13.6
DMP	12.17	14.51	19.2	13.47	7.2

Legend: μ- mean service rate (transactions/second)
p- branching probability (i.e., visit ratio)
TPUT- throughput (transactions/second)
UTIL- *utilization*
QLEN- mean queue length
WAIT- mean wait time (milliseconds)

The comparison between the parameter predictions and the actual parameters is good. This is in spite of large actual parametric changes when the swapping activity was moved (see Table 2, column entitled "% change"). The accuracy with which the model predicted the observed parameters provides validation for the modeling approach.

4. Conclusions

A modeling approach for the swapping activity in a computer system has been presented. A queuing network model is constructed, detailing the swapping devices. Necessary but unmeasured parameters are found via iteration/calibration. The number of iterations needed to find the unmeasured parameters (i.e., THE-REST service rate and the DMP) is small (e.g., less than 4 iterations were necessary in all cases considered).

Validation of the modeling approach is done using prediction. Since swapping mechanisms affect many system parameters, the interdependencies among these parameters are modeled. These interdependencies are modeled in various ways: 1) hardware formula calculation (e.g., the drum service rate is a function of the access time and the transfer time, which is a function of the average transaction size, which is a function of the file assignment/branching probabilities, which depend upon the swapping activity), 2) empirical formula calculation (e.g., the amount of swapping activity depends upon the number of jobs contending for resources, DMP, and is found by observing previous performance), and 3) iteration (e.g., since many parameters interrelate, guess one parameter value, say, the amount of swapping activity, calculate the other interrelated parameters, repredict the amount of swapping activity, and iterate).

The application of models of swapping provides practical benefits. The practical benefit suggested in this paper is that of tuning. That is, which set of devices and which loading factors for each device should be used for swapping to obtain optimal performance? Another application is the determination of the optimal number of interactive terminals to be active at any one time. If too few terminals are connected, the resources are underutilized. If too many terminals are connected, thrashing results. Since the model considers the effects of swapping, the point where the addition of another terminal would cause more extra swapping than extra resource utilization can be determined.

These applications, accurate models of parallelism and blocking, and the discovery of parameter interdependencies are subjects for further research.

ACKNOWLEDGMENTS

We are grateful for the sugestions and contributions made by the Systems Design and Analysis Group and the Systems Staff of the Computer Science Center at the University of Maryland.

References

[Bard 80] Y. Bard "A model of shared DASD and multipathing," CACM 23, 10 (October 1980), 564-572.

[BB 80] S.C. Bruell and G. Balbo Computational Algorithms for Closed Queueing Networks, North Holland, 1980.

[BBC 77] R.M. Brown, J.C. Browne, and K.M. Chandy "Memory management and response time," CACM 20, 3 (March 1977), 153-165.

[BCMP 75] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios-Gomez "Open, closed and mixed networks of queues with different classes of customers," JACM 22, 2 (April 1975), 248-260.

[CHW 75] K.M. Chandy, U. Herzog, and L. Woo "Parametric analysis of queuing networks," IBM Journal of Research and Development 19, (January 1975), 36-42.

[CN 80] K.M. Chandy and D. Neuse "Fast accurate heuristic algorithms for queueing networks models of computing systems," University of Texas Technical Report TR-157, Department of Computer Sciences, Austin, Texas (September 1980).

[Comp 80] Computer, Special Issue: Analytical Queueing Models, 13, 4 (April 1980).

[CS 78] Computing Surveys, Special Issue: Queueing Network Models of Computer System Performance, 10, 3 (September 1978).

[CS 80] K.M. Chandy and C.H. Sauer "Computational algorithms for product form queueing networks," CACM 23, 10 (October 1980), 573-583.

[DAGT 79] L.W. Dowdy, A.K. Agrawala, K.D. Gordon, and S.K. Tripathi "Computer performance prediction via analytical modeling -- an experiment," Proceedings of the Conference on Simulation, Measurement, and Modeling of Computer Systems, (August 1979), 13-18.

[Hunt 81] D. Hunter "Modeling real DASD configurations," presented at the ORSA-TIMS Special Interest Meeting: Applied Probability - Computer Science, the Interface (January 1981).

[DB 78] P.J. Denning and J.P. Buzen "The operational analysis of queueing network models," Computing Surveys 10, 3 (September 1978), 225-261.

[KKT 80] P.S. Kritzinger, A.E. Krzesinski, and P. Teunissen "Incorporating system overhead in queuing network models," IEEE Transactions on Software Engineering SE-6, 4 (July 1980), 381-390.

[KT 78] A. Krzesinski and P. Teunissen "A multi-class network model of a demand paging computer system," Acta Informatica 9, 1978, 331-343.

[RL 80] M. Reiser and S.S. Lavenberg "Mean value analysis of closed multichain queueing networks," JACM 27, 2 (April 1980), 313-322.

[SB 80] C. Smith and J.C. Browne "Aspects of software design analysis: concurrency and blocking," Proceedings of Performance '80 International Symposium on Computer Performance Modelling, Measurement and Evaluation (May 1980), 245-253.

[SM 79] C.H. Sauer and E.A. MacNair "Queueing network software for systems modeling," Software Practice Experience 9, 5 (May 1979), 369-380.

[Trip 79] S.K. Tripathi "On approximate solution techniques for queueing network models of computer systems," Ph.D. Thesis, University of Toronto, 1979.

[Zaho 80] J. Zahorjan "The approximate solution of large queueing network models," Ph.D. Thesis, University of Toronto, 1980.

[ZHS 78] J. Zahorjan, J.N.P. Hume, and K.C. Sevcik "A queueing model of a rotational position sensing disk system," INFOR 10 (1978), 199-216.