Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

USING SIMULATION TO EVALUATE ANALYTIC MODELS OF MEMORY QUEUEING

Charles E. Knadler, Jr.
and
Ralph M. May

IBM Federal Systems Company
9201 Corporate Boulevard
Rockville, Maryland 20850

ABSTRACT

Discrete event simulation is used to evaluate analytic models of memory constrained central server systems. Flow equivalent (load dependent) service centers are introduced and used regain separability and analytical tractability. The analytic models' accuracy is shown to improve with increasing population size.

The method of surrogates is also discussed. A simple response time criterion illustrates when that method can be used to improve on the flow equivalent service center approach and when it will necessarily at best provide equal accuracy.

1 INTRODUCTION

Simulation is used to study the accuracy of queueing network models of memory constrained computer systems. Queueing network theory grew out of the realization by Buzen (1976) and Denning and Buzen (1978) that certain queueing system equations could be derived from operational variables and did not depend on the restrictive assumptions of Markovian queueing theory. However the simple product form solutions (Reiser and Lavenberg 1980, Lazowska et al 1984) require that the models are separable. Separability requires:

- Flow balance at service centers: the number of job arrivals equals the number of completions.
- One step behavior: no two jobs change state at exactly the same time.
- Homogeneous routing: job routing between centers is independent of the queue sizes at the centers.
- Homogeneity of devices: job completion behavior at a center is independent of the placement of jobs at other centers.
- Homogeneous arrivals: the arrival times of jobs external to the network are independent of the placement and number of jobs in the network.

Queueing network analysis is probably unequalled for its ability to provide mean value performance measures accurate to on the order of 10 to 20% for little effort other than model parameterization. This type of analysis has been used successfully to select, configure and tune computer systems (Lazowska et al 1984, Lipsky and Church 1977, Zahorjan et al 1982). It is particularly useful for sensitivity analysis.

In earlier work, queueing network models have been shown to be robust for single class central server models (Knadler 1991).

(1) Simultaneous resource possession. The analytical models provide good estimates of throughput and residence time with as many as 25% of all interactive jobs queueing for memory.

(2) Load dependent service demand. The mean value analysis provides excellent results for disk systems over a broad range of channel utilizations, using a simple iterative extension.

(3) Independently distributed service times. Excellent agreement is obtained even though disk service times are not independent of each other, as a result of channel contention.

(4) Exponentially distributed service times. Simulation shows central server performance measures to be relatively constant (within 15%) as a function of the coefficient of variation of central processor services times, for coefficients of variation in the range (0.25,8.0).

One area attracting interest, which is not fully addressed in Knadler (1991) is simultaneous resource possession, or passive resource contention (Jacobson and Lazowska 1982, Lavenberg 1989). Memory constraints are an example of passive resource contention. This example is of interest because memory constraints can both reduce the system throughput and increase system response time. The simultaneous nature of memory possession also results in nonseparable queueing

networks.

In extending the models to memory constrained systems, the theory must be adjusted to account for the passive nature of memory contention and the associated nonseparability it introduces into the system models. One particular approach involves the use of the concept of flow equivalent service centers (FESC), an example of hierarchical modelling techniques.

Consider the interactive system model, shown in figure 1. We conceptually divide the model into two parts: the aggregate and the complement, figure 2. The complement in this case consists of the terminals and the aggregate consists of the memory constrained central server system.

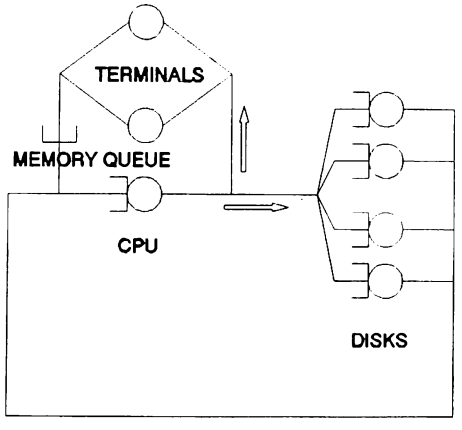

Figure 1 Simulated Interactive Computer System

In order to regain separability, we model the aggregate as a load dependent service center (see *Defining the FESC* below). This approach (Chandy et al 1975a, Chandy et al 1975b, Lazowska et al 1984) is analogous to the Norton equivalent circuit technique of linear circuit theory. Instead of circuit characteristics, we need to consider queueing system characteristics; e.g., residence time, arrival and departure patterns, or queueing disciplines. The characteristics of the FESC are chosen so that its average throughput and average residence time are the same as those in the memory constrained central server system. Unlike the original model, the new model, figure 3, is separable. The explicit memory queue has been eliminated and its impact on performance incorporated indirectly through load (queue length) dependent service rates in the FESC.

2 APPROACH

2.1 The Simulation

A discrete event simulation of a multitasked

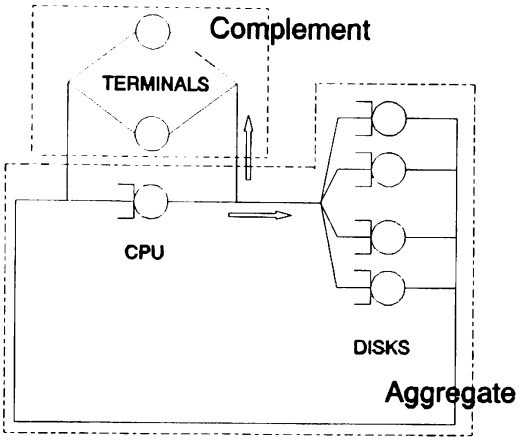

Figure 2 Central Server System: Complement and Aggregate

computer system is used to study the accuracy of network queueing analysis. The system modelled is the interactive computer system shown in figure 1. The disk subsystem uses rotational position sensing (rps). Tasks are characterized by the number of visits to the cpu and disk subsystems, their service demands, and main memory requirements. The simulation is based on a disk input/output model described in Knadler (1991) and is implemented using P_SMPL, a Pascal implementation of the SMPL system (MacDougall 1987).

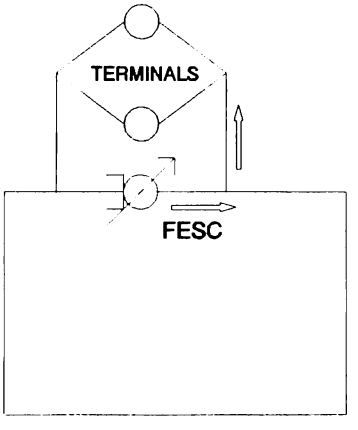

Figure 3 Central Server FESC

The interactive system is a memory constrained central processor attached to a network of interactive users and an input/output system. Processor tasks have exponentially distributed service time per visit. The number of visits per task and task memory requirements are uniformly distributed. Disk data requirements are expressed as a uniformly distributed number of sectors requested per access.

A task, a terminal system request, requests memory

when it enters the system and holds it until it completes all visits to both the central processor and the disks. This mechanism matches either the allocation of virtual memory in a paged system (queueing is for virtual memory, not real memory) and the allocation of real memory in a segmented system.

2.2 Defining the FESC

The FESC model of the central server system is constructed using mean value analysis (Lazowska et al 1984, Knadler 1991) to determine its load dependent service rates. One starts by establishing the average multiprogramming level of the system. This is usually calculated as the ratio of total available memory to the average task memory requirement.

Ignoring terminal think times, for the moment, one builds a batch processing model of the system. This batch system has the derived multiprogramming level and job characteristics of the original system. This batch load model is used to establish system throughput limits.

The CPU and attached disks are replaced with a single FESC whose throughput is defined by

$$\mu(n) = \begin{cases} X(n), & n \le N \\ X(N), & n > N \end{cases}$$

Where μ(n) and X(n) are the FESC and batch throughput, respectively, for systems with a population of n and N is the multiprogramming level.

For the central server model, the equations used to calculate system parameters can be simplified as follows. Define p(i | j) to be the percentage of time the central server has i customers when there are j customers in the entire system. Starting with n = 0 and iterating until n = maximum number of customers, one successively does the following set of three calculations. R(n), the average residence time with n customers in the system, is defined by

$$R(j) = \sum_{i=1}^{j} \frac{iV}{\mu(i)} p(i-1 \mid j-1)$$

For a given population, j, the FESC throughput is

$$X(j) = \frac{j}{Z+R(j)}$$

The terms p(i | j) can be calculated as

$$p(i \mid j) = \frac{X(j)}{\mu(i)} p(i-1 \mid j-1) \quad i=1,\ldots j$$

$$p(0 \mid j) = 1 - \sum_{r=1}^{j} p(r \mid j);$$

$$q(j) = \sum_{r=1}^{j} r.p(r \mid j)$$

This approach is a variation on the standard iterative calculations associated with mean value analysis (MVA). The changes are designed to account for the load-dependent nature of FESC behavior.

The first step in constructing the FESC is the calculation of the system throughput rates for a given multiprogramming level. Integer values that bound the calculated multiprogramming level are used. For example, if the multiprogramming level is 23.822, then queueing models are run with levels set at 23 and 24. The average CPU demand is set to match the simulation average of 64 milliseconds. The disk service demand values are set equal to the values determined in simulation. This approach was taken in an attempt to eliminate the effect of parameterization errors on the accuracy of the FESC approach.

Using the FESC queueing results for integer multiprogramming levels, weighted linear averages of the queueing results are computed in order to approximate the simulation multiprogramming levels. These results are then compared to the simulation results.

3 RESULTS

Analysis of the simulation results reveals a subtlety in the calculation of average multiprogramming level described above. If the average memory requirement is considered to be the average over all jobs, an error will be introduced. What is needed is the time averaged multiprogramming level and this is biased by the number of jobs concurrently in memory (the instantaneous multiprogramming level). Queue length is a nonlinear function of the instantaneous multiprogramming level. Thus jobs with small memory requirements will tend to have greater queueing delays than jobs with large memory requirements, since there will typically be a larger instantaneous multiprogramming level in the first case than in the second. This effect is illustrated in figure 4, where the average multiprogramming level is compared with the calculated ratio of available memory to average memory requirement. The correct time average value of the multiprogramming level is used for the FESC, again to eliminate parameterization error bias.

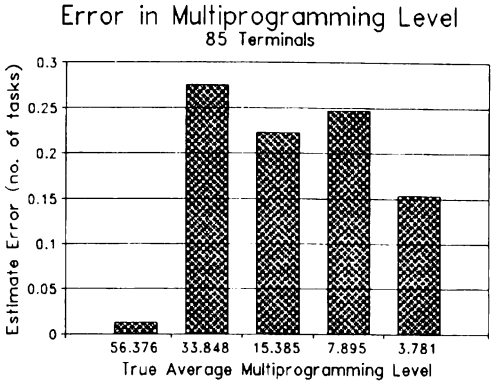

Figure 4 Underestimate of Multiprogramming Level

If the memory constraint is on either nonpaged real memory or on virtual memory, memory utilization must be considered or the average multiprogramming level will be over estimated. The ratio, of available memory to average memory requirement, must be multiplied by the average memory utilization to account for the fact that enough memory must be available to satisfy the jobs total memory requirement.

Using the correct multiprogramming level, the central server produces excellent agreement with simulation results (figure 5). Throughput agrees to within 8.5% over the entire range and response time matches to within 12.7%. In addition note that the FESC has very good relative behavior; i.e., tends to be better in predicting relative change than absolute change. Therefore it is an excellent tool for tradeoff studies and initial system performance estimates.

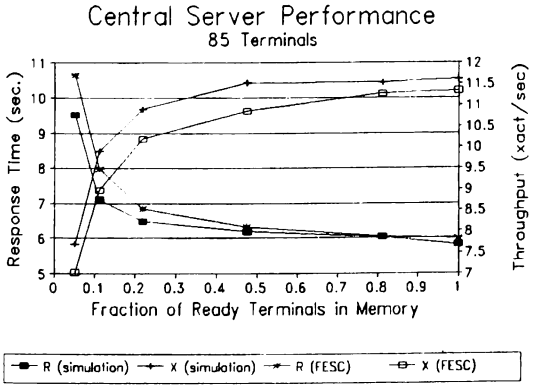

Figure 5 Central Server System Performance

Figures 6 and 7 illustrate the effects of population size on the FESC central server model's accuracy. The percentage accuracy of the throughput and residence time estimates decreases somewhat with decreasing population size. Maximum percentage errors in throughput increase to 12.1% and 12.4%, while the errors in residence time increased to 21% and 45%, respectively for the 45 and 20 terminal cases.

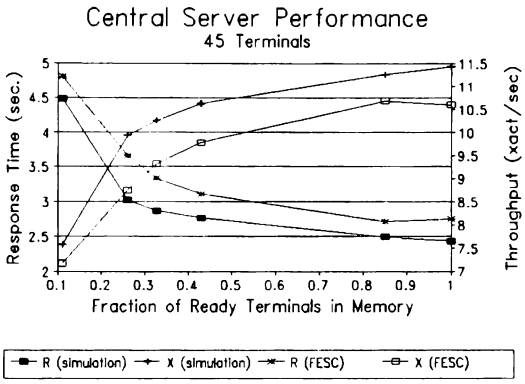

Figure 6 Central Server Performance

There is an implicit assumption in the FESC model, that its output rate is a function of only the number of customers in the aggregate. This will only be true if the aggregate is in local equilibrium (Lazowska et al 1984). This will be more likely with the larger populations than with the smaller populations. Thus the increasing accuracy with increasing multiprogramming level is not surprising.

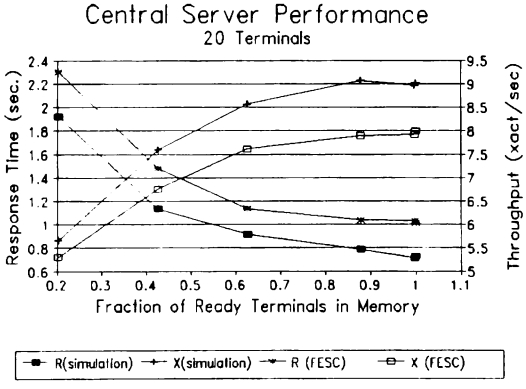

Figure 7 Central Server Performance

4 METHOD OF SURROGATES

The method of surrogates, an alternative approach for handling simultaneous resource possession, has been developed by Jacobson and Lazowska (1982). The shared resources are partitioned into two sets: primary resources and secondary resources. The set of resources acquired first are called primary resources and the set acquired later are called secondary resources. Thus if a

part of a shared resource's holding time is not overlapped with another, it is considered a primary resource. While if all a resource's holding time is overlapped with another shared resource, it is considered a secondary resource.

Consider the memory constrained, interactive computer system of figure 1, the memory is the primary resource and the central processor and input/output system are the secondary resources. Service demands are also partitioned into non-overlapping components, not necessarily related to physical queues. The primary partition includes the queueing time which can only be reduced by changes to the primary resource (memory capacity in this case) and the secondary partition includes the queueing time which can be impacted by changes to the secondary resource (central processor or input/output system speed).

The primary partition's service demand is zero (there is only waiting time associated with the memory queue, no service time) and the secondary partition's service demand is the sum of the cpu and the input/output subsystem service demand. This zero service demand of the memory resource is the motivation for referring to this form of simultaneous resource possession as passive

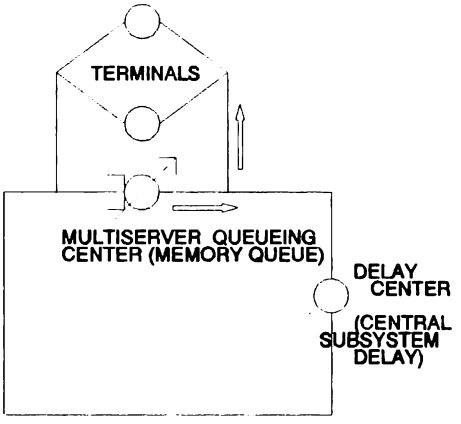

Figure 8 Method of Surrogates: Model 1

resource contention.

Two models are formed, figures 8 and 9. In the first the primary system, memory, is modelled as a multiserver queueing center and the secondary subsystem, processor and input/output, is modelled as a delay center. The number of servers is the number of memory partitions (multiprogramming level) and the service demand, $D_{1,MEMORY}$, is the same as the sum of the service demand at the cpu and at each disk. While in the second model, the central subsystem is modelled by the same FESC discussed previously, figure 3, and the memory resource is modelled as a delay center.

The two models are solved iteratively. On the first iteration, the service demand for model one's delay

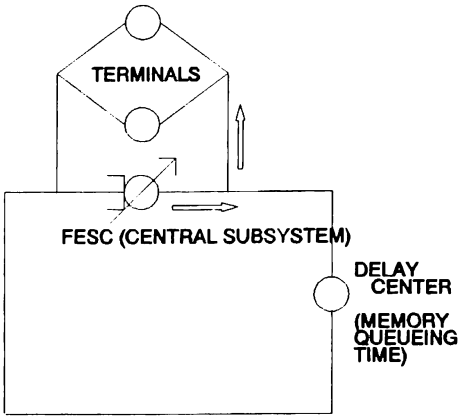

Figure 9 Method of Surrogates: Model 2

center is set to zero and the model is solved. The waiting time calculated for the multiserver, memory queueing time, is input into the second model as the service demand for the second model's delay center. The second model is solved and the difference between the FESC's residence time and $D_{1,MEMORY}$ is input into the first model as the delay center's service demand. This process is repeated until the same memory queueing time is calculated for two successive iterations.

While this technique has broad applicability, it will not improve on the accuracy of the simple FESC approach discussed earlier for the cases under consideration. As is seen in figures 5, 6, and 7; the FESC's response time is always greater than the true response time. Thus if, after iterating, a zero memory queueing delay is found for model one the accuracy will be the same. With nonzero memory queueing delay, the system response time will be greater than the simple FESC and thus less accurate. In matter of fact, the Jacobson and Lazowska technique calculated very small memory delays and the accuracy was almost equal to the simple FESC model.

5 SUMMARY AND CONCLUSIONS

FESC models of single class central server systems have been shown to be sufficiently accurate to be used for tradeoff and sensitivity studies of memory constrained systems. However, several weakness of the analytical approach are also apparent.

- Simulation or measurement is required to obtain good estimates of the multiprogramming levels and disk service times.
- The single class models execute much faster than the simulation of the equivalent system, but analytical multiclass models often require much

greater computer resources than does simulation.

The analytical results have been compared to time-averaged results for 200 seconds of simulated system performance and not to confidence intervals. Experience has shown this simulation interval will produce statistically meaningful results for central service systems. This is illustrated in figure 10, which compares the 200 second average value, the analytical results, and the confidence intervals for the 85 terminal cases considered above. The simulation results are seen to be virtually identical to the upper bound of the 90% confidence interval. Multiple runs and ensemble averages will normally be required, however, increasing the relative cost of simulation to analytic solutions.

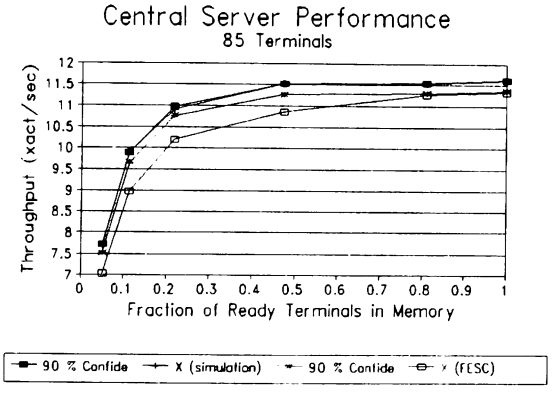

Figure 10 90% Confidence Interval

There is an ongoing debate between the use of simulation and queueing network models, possibly missing an important point. They can be used to supplement each other. Based on the authors' experience, the major cost of a queueing network study is parameterization. With a properly designed effort, this parameterization could be a virtually free result of a simulation effort. The analytical models could then be used both to help verify the simulation model and for the many "what-if" studies desired, but often too expensive to perform using simulation.

As discussed earlier, queueing network analysis is probably unequalled for its ability to provide mean value performance measures accurate to on the order of 10 to 20% for little effort other than model parameterization. This type of analysis has been used successfully to select, configure and tune computer systems (Lazowska et al 1984, Lipsky and Church 1977, Zahorjan et al 1982). It is particularly useful for sensitivity analysis. However, its strengths are also its weaknesses. The algorithms discussed provide mean values with little insight into variation of performance measures. If it is important to study confidence limits as well as mean values of performance measures, then the use of simulation is indicated. The simulation runs can produce a wealth of statistics for all aspects of system performance, in addition to the mean values of throughput, residence time, queue lengths, and service center utilization provided by the analytical models.

One of the negatives of the use of simulation, development cost, can be mitigated by good methodology. Simulations should be developed only after a good understanding is gained of what questions are to be answered (Balci 1990). The level of detail used in a simulation should be chosen to be sufficient to answer these questions and avoid unnecessary effort. Experience indicates that the simulation value to cost ratio should be expected to be low for simulations developed without a careful examination of the questions to be answered and the best techniques to be used to arrive at these answers. While the simulation value to cost ratio can be quite high when techniques suggested by (Balci 1990) are used. Relatively simple simulations have provided good and sufficient insights into system performance; e.g. (Mitchell et al 1974) and (Knadler and May 1990). However to get statistically meaningful results, multiple simulation runs and statistical analysis of results are required (Law 1990, Law and Kelton 1991) and for complex systems these can take lots of resources.

6 FURTHER WORK

Due to the intractability of the exact load dependent and nonload dependent multiclass algorithms, for even moderately sized populations and a moderate number of classes, approximate solution techniques are required. Two techniques are available for nonload dependent service centers (Lavenberg 1989 and Lazowska et al 1984), but no approximate solution technique has yet been developed for the load dependent service centers.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their comments on a draft of this paper.

REFERENCES

Osman Balci, 1990. Guidelines for Successful Simulation Studies. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci et al. Association for Computing Machinery. New Orleans, Louisiana.

Jeffery P. Buzen, 1976, "Fundamental Operational Laws

of Computer System Performance, *Acta Informatica* 7,2. pp 167–182

K. M. Chandy et al, 1975a, "Parametric Analysis of Queueing Networks," *IBM Journal of Research and Development*, Vol. 19, No. 1, January, pp 36–42

K. M. Chandy et al, 1975b, "Approximate Analysis of General Queueing Networks," *IBM Journal of Research and Development*, Vol. 19, No. 1, January, pp 43–49

Peter J. Denning and Jeffrey P. Buzen, 1978, "The Operational Analysis of Queueing Network Models, *Computing Surveys*, Vol. 10, No. 3, pp 225–261

Patricia A. Jacobson and Edward D. Lazowska, 1982, "Analyzing Queueing Networks with Simultaneous Resource Possession," *Communications of the ACM*, Vol. 25, No.2, February, pp 142–151

Charles E. Knadler and Ralph May. 1990. Disk I/O, a Study in Shifting Bottlenecks. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci et al. 826–830. Association for Computing Machinery. New Orleans, Louisiana.

Charles E. Knadler, Jr., 1991, "The Robustness of Separable Queueing Network Models," in *Proceedings of the 1991 Winter Simulation Conference*, eds. Barry L. Nelson et al, Association for Computing Machinery, Phoenix, Arizona, pp 661–668

Stephen S. Lavenberg, 1989, "A Perspective on Queueing Models of Computer Performance," *Performance Evaluation*, Vol. 10, pp 53–76.

Averill M. Law, 1990. Design and Analysis of Simulation Experiments for Manufacturing Applications. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci et al. 33–37. Association for Computing Machinery. New Orleans, Louisiana.

Averill M. Law and W. David Kelton, 1991, *Simulation Modeling and Analysis, Second Edition*. New York: McGraw–Hill.

Edward D. Lazowska et al, 1984, *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*. Prentice Hall, Englewood Cliffs, NJ

L. Lipsky and J. D. Church, 1977, "Applications of a Queueing Network Model for a Computer System," In *Computing Surveys*, Vol. 9, No. 3 (September), pp 205–221

M.H. MacDougall, 1987, *Simulating Computer Systems: Techniques and Tools*. The MIT Press, Cambridge, MA

John Mitchell et al, 1974, Multiprocessor performance analysis. In *Proceedings of the 1974 National Computer Conference*. 399–403.

M. Reiser and S.S. Lavenberg, 1980, "Mean Value Analysis of Closed Multichain Queueing Networks," *Journal of the ACM*, Vol. 27, No. 2, April, pp 313–322

Charles H. Sauer, 1981, "Approximate Solutions of Queueing Networks with Simultaneous Resource Possession," *IBM Journal of Research and Development*, Vol. 25, No. 6, November, pp 894–903.

J. Zahorjan et al, 1982, "Balanced Job Bound Analysis of Queueing Networks," In *Communications of the ACM*, Vol. 25, No. 2 (February), pp 134–141

AUTHOR BIOGRAPHIES

CHARLES E. KNADLER, JR. is a senior engineer for the IBM Federal Systems Corporation's Advanced Automation System project. He is also an Associate Professorial Lecturer in computer science at the George Washington University and is the Program Chair for the 1993 International Conference on Simulation in Engineering Education. His research interests include computer architecture, performance analysis and the use of personal computers in computer science research.

RALPH M. MAY is an Advisory Engineer/Scientist with IBM's Federal Systems Company. He is currently assigned to IBM's Federal Aviation Administration Advanced Automation System (AAS) Project, where his work centers on various aspects of air traffic control system monitoring, measurement and performance analysis. His other assignments within IBM have involved work on software prototyping and communications systems performance analysis. Prior to joining IBM, he held positions at MCI Telecommunications and The Center for Naval Analyses. His interests include performance analysis, computer and communications system architectures, algorithms and object oriented programming.