SHARED MEMORY PARALLEL REGENERATIVE QUEUING NETWORK SIMULATION

Panajotis Katsaros
Constantine Lazos
Department of Informatics
Aristotle University of Thessaloniki
54006 Thessaloniki, Greece
E-mail: {katsaros, clazos}@csd.auth.gr

KEYWORDS
Queuing models, Performance analysis, Statistical analysis, Parallel simulation.

ABSTRACT

Discrete-event stochastic simulation is one of the most commonly used tools for performance modeling and evaluation. Parallel/distributed simulation enables a simulation program to execute on a computing system containing multiple processors and aims in reducing the model's execution time. Three basic types of execution mechanisms have appeared. The first two (the conservative and the optimistic approach) aim in partitioning the simulation model into a number of sub-models, also called logical processes (LPs). Their emphasis, lies on the specification of the appropriate synchronization, deadlock handling and/or memory management algorithms. The third approach (known as the time parallel approach or simply as Multiple Replications in Parallel Time Streams), aims in overcoming the need for sufficiently long runs in steady-state stochastic simulations, by executing multiple replications of the entire model in a parallel fashion. This work, presents a fast parallel OpenMP based implementation, for multivariate queuing network simulations. The simulation results are statistically processed, by applying the classical regenerative method under the Lavenberg & Sauer sequential analysis procedure. The first experimental results indicate significant speedups accompanied by acceptable confidence interval coverage.

1. INTRODUCTION

Parallel/distributed discrete event simulation has grown rapidly over the last years, as a result of the need to employ the high-potential processing power of the modern multiprocessors, for speeding up simulation runs. Research on this field has been primarily concentrated in developing algorithms where several processors cooperate on a single realization of the stochastic process simulated (Fujimoto 2000). However, the effectiveness of such an approach depends on the level of inherent parallelism that exists in the simulated model.

A *time parallel simulation*, partitions the simulation time into a number of non-overlapping intervals, as many as, the number of the available processors. A logical process assigned to a particular processor, computes the portion of the sample path within the corresponding time interval. The fundamental problem that has to be overcome by a time parallel simulation algorithm is to ensure that the model states computed at the "boundaries" of the time intervals, match each other.

In this sense, the *regenerative method* provides a safe way to partition the simulation time into a number of intervals. This method was independently developed in (Cox and Smith 1961; Fishman 1973; Crane and Iglehart 1974) and it is based on the detection of a recurrent model state, which can be considered to represent the system in a particular time instant, within its steady-state phase. Every time the model passes through this state, the experiment is probabilistically restarted. Thus, a single simulation run is partitioned into cycles.

Since the classical regenerative estimation depends only on the difference between the time instant of the end and the time instant of the start of a cycle (Katsaros and Lazos 2000a), we can assume parallel time streams starting from the time instant 0, as representing a single regenerative realization of the simulated model. Each time stream contains an integer number of regenerative cycles. All the regenerative observations generated by each one of the available processors, are independent to each other and they can be used in producing confidence intervals.

Time parallel regenerative algorithms have been also used in (Andradottir and Ott 1995; Fujimoto and Nikolaidis 1995; Rego and Sunderam 1992). However, this is the first work, where the method's validity is being assessed not only in respect to the succeeded speedups, but also in respect to the resulted confidence interval *coverage*. Coverage is defined (Pawlikowski et al 1998) as the relative frequency with which the confidence interval contains the true parameter.

The conducted experiments were carried out by the use of a general queuing network simulator, developed by us under the Guide C++ OpenMP compiler on a SUN E3500 shared memory system. The slow multiplicative congruential pseudorandom number generator which is usually used in similar software (Katsaros and Lazos 2000b), has been substituted by a new generalized feedback shift register (GSFR) ultra fast generator, known as the Mersenne Twister (Matsumoto and Nishimura 1998).

Another innovative aspect of this work is the multivariate nature of the conducted experiments, as opposed to the univariate evaluations presented in most similar studies. This has leaded us, to valuable conclusions and recommendations for the safe use of the suggested approach in practical multivariate studies.

Similar experimental evaluations have been presented, for the independent replications approach (Heidelberger

1986), the batch means and the non-overlapping batch means (Pawlikowski and Yau 1993) and the spectral analysis methods (Raatikainen 1992; Pawlikowski et al 1994).

2. THE REGENERATIVE METHOD FOR SIMULATION ANALYSIS

A regenerative process $\{X(t): t \geq 0\}$ with state space R^k, is (Iglehart 1978) a stochastic process which starts from scratch at an increasing sequence of regeneration times $\{\beta_i: i \geq 1\}$. That is, between any two consecutive regeneration times β_i and β_{i+1}, say, the portion $\{X(t): \beta_i \leq t < \beta_{i+1}\}$ of the regenerative process is an independent, identically distributed replicate of the portion between any other two consecutive regeneration times. However, the portion of the process between times 0 and β_1, while independent of the rest of the process, is allowed to have a different distribution. The typical situation in which the regenerative assumption is satisfied is when β_i represents the time of the ith entrance to a fixed state **s**, say, and upon hitting this state the simulation proceeds without any knowledge of its past history.

Thus, the problem takes the form of detecting such a recurrent state, which can be considered to represent the system in a particular time instant within its steady-state phase.

It has been shown (Glynn 1994) that all "well-posed" steady-state simulation problems are regenerative. Moreover, Shedler (1993) provides valuable results for identifying appropriate regenerative states in queuing network simulations. More precisely, let us assume that at any time instant, each job, is of exactly one class and one type. Jobs may change class as they traverse the network, but they cannot change type. The type of a job may influence its routing path through the network as well as its service requirements at each service center. Service priorities can also be associated with job types. Any state **s**, where all jobs, are placed at a service center, which sees only one class, or is such that, jobs of the lowest priority are subject to pre-emption, has been proved to be a regenerative state.

Let us assume, our aim is to estimate the mean value of a queuing network characteristic (e.g. throughput), which is given, as a real-valued function **f** over the regenerative stochastic process $X = \{X(t) ; t \geq 0\}$

$$k(f) = E[f(X)]$$

Let us also call

$$Z_k(f) = \int_{T_{k-1}}^{T_k} f(X(u)) \cdot du$$

the observation produced by the kth regenerative cycle. A 100 α % confidence interval for k(f), after the completion of N regenerative cycles, is given (Iglehart 1978) by

$$\left[\hat{k}(N) - \frac{s(N) \cdot F^{-1}\left(\frac{1+a}{2}\right)}{\sqrt{N} \cdot \bar{\tau}(N)}, \hat{k}(N) + \frac{s(N) \cdot F^{-1}\left(\frac{1+a}{2}\right)}{\sqrt{N} \cdot \bar{\tau}(N)} \right] \quad (1)$$

where

$\bar{\tau}(N)$ is the average cycle length

and (1)(1)

$$\hat{k}(N) = \frac{\bar{Z}(N)}{\bar{\tau}(N)} \quad (2)$$

$$s^2(N) = s_{11}^2(N) - 2\hat{k}(N) s_{12}^2(N) + (\hat{k}(N))^2 s_{22}^2(N) \quad (3)$$

with

$$s_{11}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (Z_k(f) - \bar{Z}(N))^2 ,$$

$$s_{22}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (\tau_k - \bar{\tau}(N))^2$$

$$s_{12}^2(N) = \frac{1}{N-1} \sum_{k=1}^{N} (Z_k(f) - \bar{Z}(N))(\tau_\kappa - \bar{\tau}(N))$$

However, although the estimator given in (2), is a consistent estimator, which means that it tends to the mean value with probability 1, as $N \to \infty$, it is not unbiased. Other estimators that have been suggested, in an attempt to reduce the bias introduced by the aforementioned (classical) estimator are the Fieller, the Beale, the jackknife and the Tin point estimators (Iglehart 1978). Respectively, different estimation procedures are being applied in deriving confidence intervals in the Fieller and the jackknife cases.

Another critical issue in producing confidence intervals, that cover the true steady-state mean with the desired probability level, is the way the simulation run length is determined. The reason is that different systems behave in radically different ways and thus require radically different run lengths to generate adequate confidence intervals. Thus, no procedure in which the run length is fixed, before the simulation begins, can guarantee accurate results. Instead of this, sequential procedures, which determine the length of the simulation during the course of the run, are preferred. At least two different sequential procedures have been suggested for use in regenerative stochastic simulations.

The Lavenberg and Sauer (Lavenberg and Sauer 1977) sequential procedure, which has been used in this work, is a direct consequence of (1) and determines the number N of the required regenerative cycles as

$$N \geq \left(\frac{F^{-1}\left(\frac{1+a}{2}\right)}{\delta} \right)^2 \cdot \left(\frac{s(l)}{\hat{k}(l) \cdot \bar{\tau}(l)} \right)^2 \quad (4)$$

where $s(l), \hat{k}(l), \bar{\tau}(l)$ are the sample estimates, after the l th cycle of the simulation experiment. Thus, this approach assumes that the required number of cycles in order to achieve the desired accuracy is recalculated at the end of each cycle.

An interesting comparative study and survey of sequential procedures for steady-state (serial) simulations, can be found in (Law and Kelton 1982).

In conclusion, although the regenerative method is not the only one used for generating confidence intervals, it is a highly desirable approach since it possesses valuable asymptotic properties and a sound theoretical basis. Moreover, it is the only one method that retains the possibility of implementing algorithms, for obtaining derivatives of expectations with respect to various parameters (Reiman and Weiss 1986; Glynn 1987). It allows performing model sensitivity studies and optimization based on the results of a single simulation run.

A thorough survey of all the available methods for simulation output data analysis and the problems associated with applying them in queuing models is given in (Pawlikowski 1990).

3. THE PARALLEL REGENERATIVE SIMULATION

In our parallel regenerative queuing network simulator, each LP contains the entire model. A number of LPs start the execution of the experiment simultaneously. The simulation stops when the statistics produced by the sum of the regenerative cycles completed by each one of the LPs, satisfies (4) for all the performance measures of interest.

The software features an object-oriented structure which:

- Allows simulation of open or closed queuing networks with probabilistic job routing with multiple job types, different queuing disciplines, passive resources and primitive synchronization characteristics (job fission and fusion).
- Allows concurrent estimation of multiple performance measures (multivariate experiments) such as mean throughputs, mean utilizations, mean response times and mean queue lengths.
- Allows easy incorporation of new functionality and more performance measures of interest.

The software was developed under the Guide C++ OpenMP compiler on a SUN E3500 shared memory system, in Edinburgh's Parallel Computing Centre. OpenMP (1997), is the proposed industry standard Application Program Interface (API) for shared memory programming. It is based on a combination of compiler directives, library routines and environment variables that can be used to specify shared memory parallelism in C++ or Fortran.

At the moment, only the classical regenerative method and the Lavenberg & Sauer sequential procedure have been implemented. A minimum number of cycles had to be specified for each one of the performance measures of interest, since (4) was quite often temporarily satisfied after a very small number of cycles and this lead to highly inaccurate results. For the purposes of our evaluation study, 16 cycles was found to be enough, even for confidence intervals with half width less than or equal to 2% of the estimated value.

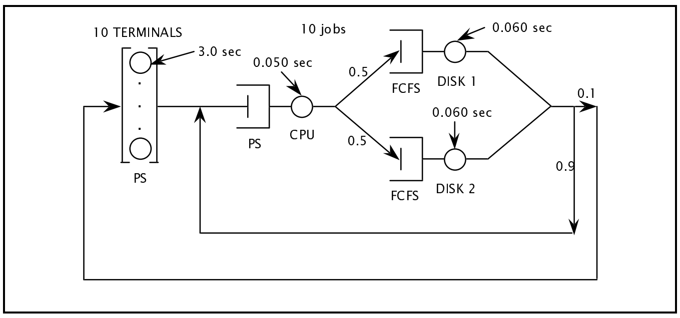

Figure 1 A Central Server Model with Terminals

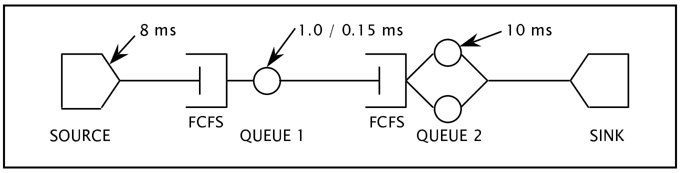

Figure 2 An Open Queuing Network Model

The conducted experiments aimed for evaluating the method's speedup gains and the resulted coverage in respect to the number of processors and for providing guidelines for the effective use of it. The queuing network models used (Figures 1 and 2) were a closed and an open network taken from (Sauer and Chandy 1981).

The performance measures of interest were the mean throughputs, mean utilizations, mean response times and mean queue lengths of the resources shown in Figures 1 and 2. The tested half width accuracies of the 90% confidence intervals were 2%, 1% and 0.5% for the first model and 2% and 1% for the second one. For each case, we have collected 200 observations with 1, 2, 4, 6 and 8 processors by differentiating the initial seed parameter in a uniform way.

Figures 3 to 6 present some of the obtained speedup and coverage (along with their 90% confidence intervals) results. We observe:

- The resulted speedup gain is dependent on the size of the model's regenerative cycle. Thus, in the case of the central server model with an average regenerative cycle of 1143 events the obtained speedups were important. In the case of the open network with an average regenerative cycle of 77 events, the time spent at the end of each cycle in the program's critical region, for checking (4), becomes important as the number of processors increases. However, in most real problems, the average cycle size is not as small as that of the simple models used in this work.
- The obtained coverage results for the Response Times in the central server model were unacceptably low, even in the case of serial execution. This can be only paid to the indirect generation of point and variance estimates through the use of the Little's formula. While the problem deserves more research, the *marked job method* (Shedler, 1993) can be used instead.
- The parallel experiments exhibit a more important coverage improvement as the relative half interval width decreases. Following Lavenberg and Sauer, a confidence interval estimation procedure can be considered to be *valid* for a particular model, if the upper endpoint of the 90% confidence interval on the true coverage is at least as large as the desired coverage, being 0.90 here. Thus, from the results obtained by parallel execution, only those obtained by using a relative half width of 1% or 0.5% for the central server and 1% for the open network can be accepted to be of an adequate quality. Moreover, their quality deteriorates as the number of processors used, increases.

Similar results have been reported in other parallel, but non-regenerative experimental studies. Moreover, in (Heidelberger 1988) a framework for the statistical analysis

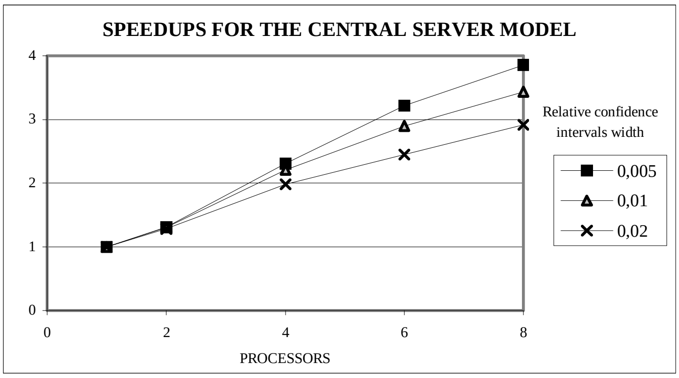

Figure 3 Speedups for the Central Server Model (200 cases)

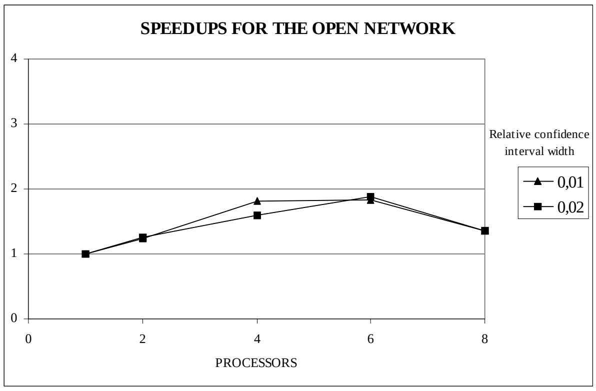

Figure 5 Speedups for the Open Network (200 cases)

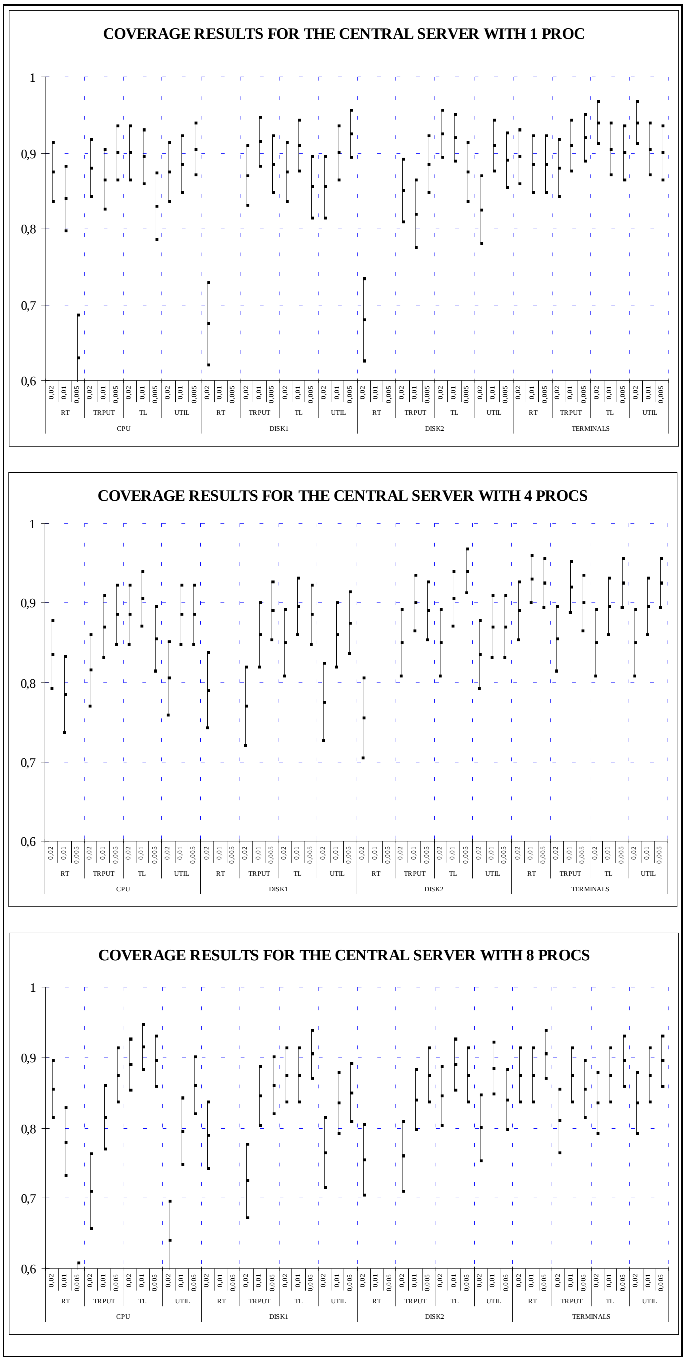

Figure 4 Coverage results for the Central Server Model with 1, 4 and 8 Processors (200 cases)

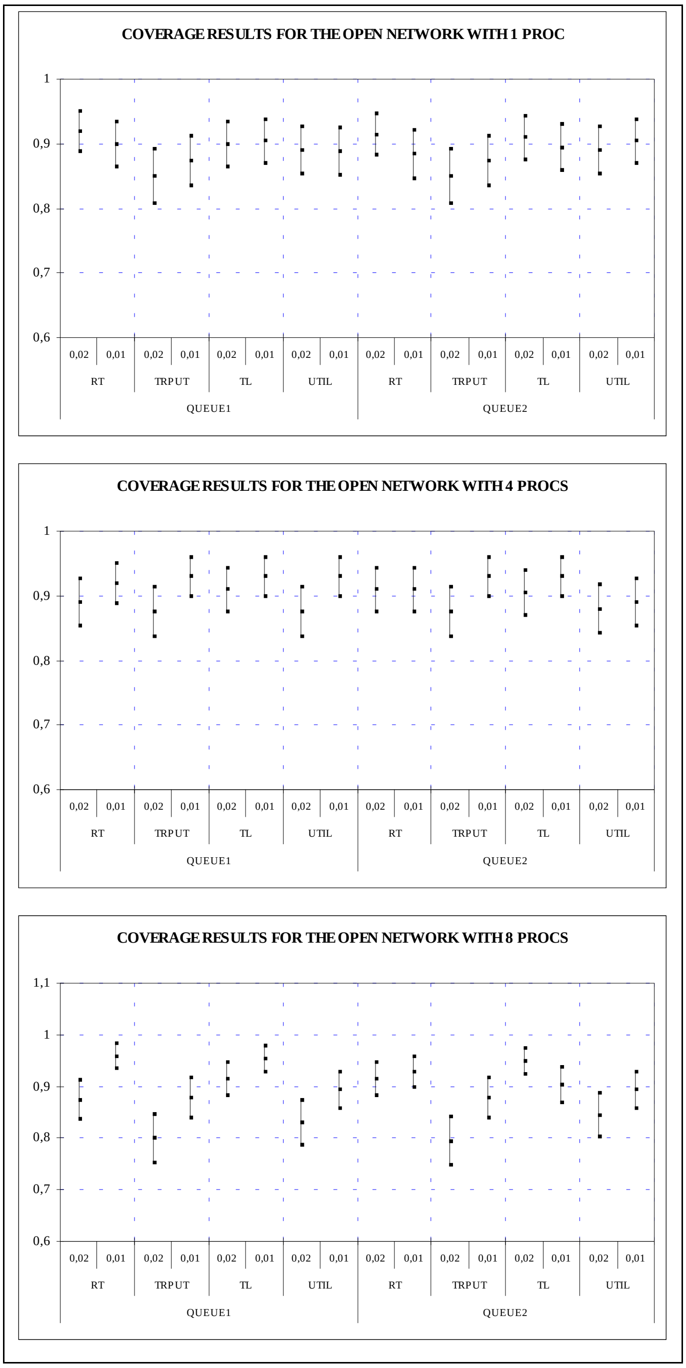

Figure 6 Coverage results for the Open Network with 1, 4 and 8 Processors (200 cases)

of parallel simulations is introduced.

4. CONCLUSION

This work presents the parallel regenerative simulation approach. The theoretical principles where this approach is based on are explained. The first experimental results show, that if our aim is to retain an adequate statistical quality, the number of processors to be used depends on the chosen precision requirements.

However, since the regenerative approach constitutes a valuable tool with applications in models' sensitivity study and optimization, the field certainly deserves more research. Other estimators and sequential analysis procedures have to be implemented and to be compared to the current classical approach.

ACKNOWLEDGMENT
The authors would like to acknowledge the support of the European Commission through grant number HPRI-1999-CT-00026 (the TRACS Programme at EPCC).

REFERENCES

Andradottir, S. and Ott, T. 1995. "Time-segmentation parallel simulation of networks of queues with loss or communication blocking", *ACM Transactions on Modeling and Computer Simulation*, Vol. 5, 269-305.

Cox, D. R. and Smith, S. W. L., 1961. *Queues*, Wiley, New York

Crane, M. and Iglehart, D..L. 1974. "Simulating stable stochastic systems: I. General multiserver queues", *Journal of ACM*, Vol. 21, 103-113.

Fishman, G. S. 1973. "Statistical analysis for queuing simulation", *Management Science*, Vol. 20, 363-369.

Fujimoto, R. M. and Nikolaidids, I., et al. 1995. "Parallel simulation of statistical multiplexers", *Journal of Discrete Event Dynamic Systems*, Vol. 5, 115-140.

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*, Wiley, New York

Glynn, P. W. 1987. "Likelihood ration estimation: An overview", *Proceedings of the 1987 Winter Simulation Conference*, Society for Computer Simulation, 366-375.

Glynn, P. W. 1994. "Some topics in regenerative steady-state simulation", *Acta Applicandae Mathematicae*, Vol. 34, 225-236.

Heidelberger, P. 1986. "Statistical Analysis of Parallel Simulations", *Proceedings of the 1986 Winter Simulation Conference*, 290-295.

Heidelberger, P. 1988. "Discrete Event Simulations and Parallel Processing: Statistical Properties", *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, 1114-1132.

Iglehart, D. L. 1978. "The regenerative method for simulation analysis", In *Current Trends in Programming Methodology*. Vol. III, Software Modeling, K. M. Chandy and P. T. Yeh, Eds., Prentice Hall, Englewwod Cliffs, N. J., 52-71.

Katsaros, P. and Lazos C. 2000. "Regenerative queuing network distributed simulation", *Proceedings of the 14th European Simulation Multiconference*, Society for Computer Simulation, 109-113.

Katsaros, P. and Lazos C. 2000. "A technique for determining queuing network simulation length based on desired accuracy", *International Journal of Computer Systems Science & Engineering*, Vol. 15, 399-404.

Lavenberg, S. S. and Sauer, C. H. 1977. "Sequential stopping rules for the regenerative method of simulation", *IBM Journal of Research and Development*, Vol. 21, 545-558.

Law, A. M. and Kelton, W. D. 1982. "Confidence intervals for steady-state simulations, II: A survey of sequential procedures", *Management Science*, Vol. 28, 550-562.

Matsumoto, M. and Nishimura, T. 1998. "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", *ACM Transactions on Modeling and Computer Simulation*, Vol. 8, 3-30.

Pawlikowski, K. 1990. "Steady-state simulation of queuing processes: A survey of problems and solutions", *ACM Computing Surveys*, Vol. 22, 123-170.

Pawlikowski, K. and Yau, V. 1993. *Methodology for stochastic simulation for performance evaluation of data communication networks*, Telecom Corporation of New Zealand, Wellington, New Zealand

Pawlikowski, K., Yau, V. W. C. and McNickle, D. 1994. "Distributed stochastic discrete-event simulation in parallel time streams", *Proceedings of the 1994 Winter Simulation Conference*, Society for Computer Simulation, 723-730.

Pawlikowski, K., McNickle, D. C. and Ewing, G. 1998. "Coverage of confidence intervals in sequential steady-state simulation", *Simulation Practice and Theory*, Vol. 6, 255-267.

Raatikainen, K. 1992. "Run Length Control using Parallel Spectral Methods", *Proceedings of the 1992 Winter Simulation Conference*, Arlington

Rego, V. J. and Sunderam, V.S. 1992. "Experiments in concurrent stochastic simulation: the EcliPSe paradigm", *Journal of Parallel and Distributed Computing*, Vol. 14, 66-84.

Reiman, M. and Weiss, A. 1986. "Sensitivity analysis via likelihood ratios", *Proceedings of the 1986 Winter Simulation Conference*, Society for Computer Simulation, 285-289.

Sauer, C. H. and Chandy, K. M. 1981. *Computer Systems Performance Modeling*, Prentice-Hall, New Jersey

Shedler, G. S. 1993. *Regenerative Stochastic Simulation*, Academic Press, Boston

The OpenMP Architecture Review Board. 1997. *OpenMP: A Proposed Industry Standard API for Shared Memory Programming*, White Paper

AUTHOR BIOGRAPHY

PANAJOTIS KATSAROS received a BSc degree in mathematics from the Aristotle University of Thessaloniki – Greece, in 1992. He also received an MSc degree in software engineering from the University of Aston in Birmingham – UK, in 1993. He is currently a PhD candidate in the Department of Informatics of the Aristotle University of Thessaloniki – Greece and he is supervised by Professor Lazos. His research interests include queuing network based performance analysis and simulation of distributed object systems and advanced sensitivity analysis techniques.

CONSTANTINE LAZOS received a BSc degree in mathematics from the University of Athens, an MSc degree in Computer Science from the University of Birmingham and a P.h.d. in Computer Science from the University of Southampton. He was a lecturer for two academic years at the University of Birmingham and he is a Professor of Computer Science at the Aristotle University of Thessaloniki, since 1980. His research interests include computer systems performance analysis and simulation techniques.