Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

STATISTICAL ANALYSIS OF PARALLEL SIMULATIONS

Philip Heidelberger
IBM Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598

ABSTRACT

This paper addresses statistical issues that arise when discrete event simulations are run on parallel processing computers. First, the statistical properties of estimates obtained by parallel distributed simulation and parallel independent replications are compared. This comparison shows that, when estimating steady state quantities, the run length, the strength of the initial transient and the asymptotic variance must be taken into account in addition to the parallel processing speed-up and the number of processors in order to determine which method is statistically more efficient. Second, the statistical properties of estimators of transient quantities obtained by the method of parallel independent replications are considered. The analysis shows that strongly consistent estimates are not obtained in finite expected time as the number of processors increases unless the computational time to complete a single replication is bounded.

1. INTRODUCTION

Parallel processing is now an active area of research as interconnected arrays of microprocessors are becoming available both commercially and in research laboratories. Because discrete event simulations often require large amounts of computing resources, they represent an important potential application for parallel processing. Chandy and Misra (1981), Jefferson and Sowizral (1985), Comfort (1983) and (1984) and Wyatt, Sheppard and Young (1983) have described different approaches to distributing a discrete event simulation onto multiple processors. A key issue for such parallel distributed simulations is to determine the amount of parallelism that can be effectively exploited.

The usual measures of a parallel computation's effectiveness are the speed-up and the efficiency. If a given job executes in time t_1 on a single processor and in time t_p on p processors, then the speed-up is defined by $\alpha_p = t_1/t_p$ and the efficiency is $e_p = \alpha_p/p$. Because of the overheads involved in such parallel computations, typically $e_p < 1$ or, equivalently, $\alpha_p < p$. Thus e_p can be thought of as the effective utilization of a processor, i.e., it is the fraction of time a processor spends doing useful work.

In parallel simulations, the efficiency is determined by many factors including:

1. The control algorithm used to ensure that events appear to processed in the correct sequence.
2. The amount of overhead required for interprocessor and interprocess communications.
3. The computer hardware and software architecture.
4. The characteristics of the system being simulated.

Idealized simulations of the Chandy and Misra null message algorithm showed near perfect efficiency for tandem queueing systems, efficiencies ranging from 0.36 to 0.62 for feedforward queueing systems, but efficiencies ranging from 0.3 to as low as 0.01 for queueing systems with feedback (Lakshmi (1979)). Similar idealized simulations of Jefferson's Time Warp algorithm yielded efficiencies ranging from 0.15 to 0.36 (Berry and Jefferson (1985)). By functionally partitioning the simulation onto four processors, Comfort (1983) reported efficiencies of about 0.45. Studies in Comfort (1984) indicate that a speed-up of between 1.2 to 1.3 can be attained by using 3 processors in parallel to manage the future event list.

Suppose a multiprocessor with P processors is available for running simulations. An alternative approach to having the P processors cooperate on a single realization of the simulation is to run, in parallel, one independent replication of the model on each processor thereby obtaining P iid (independent and identically distributed) estimates that can be averaged together. Assuming that the processors do not interfere with each other (e.g., assuming each processor has sufficient memory to run the model), the only synchronization overhead is in loading the model and simulator into each processor at the beginning of the runs and averaging the results together at the end of the runs. This overhead should be negligible, resulting in near perfect efficiency.

Which of these two approaches is better? In this paper, a simple model will be formulated that provides insight into this question. While not answering the question for any particular system to be simulated, control algorithm, or hardware and

software architecture, the model shows what the trade-offs are and identifies the key parameters that determine which approach is *statistically* more efficient in the sense of producing estimates with a smaller mean squared error for a given amount of computing resources. The model shows that a new set of factors must be considered in addition to the usual speed-up or efficiency measures. These factors include the extent of the initial transient and the inherent variability of the system being simulated, the amount of computing time available and the number of processors.

In Section 2, the model will be formulated and the trade-offs between the two approaches will be illustrated. In Section 3, the model will be extended to permit an optimization of a combination of independent replications and distributed simulation. These models show that it is often advantageous to run independent replications and in Section 4 some surprising results are presented concerning the statistical properties of parallel independent replications. Finally, Section 5 summarizes the results.

2. TRADE-OFFS IN DISTRIBUTED SIMULATION

In this section a model is formulated that forms the basis for comparing parallel independent replications with a parallel distributed simulation. The model compares the mean squared errors of estimates produced by the two methods after a fixed amount of computing time.

Some notation is required. Assume that the process being simulated is $\mathbf{X} = \{X_s, s \geq 0\}$ and that this process converges to a steady state random variable X, i.e., $\lim_{s\to\infty} P\{X_s \leq x\} = P\{X \leq x\}$. The goal of the simulation is to estimate a steady state parameter $\mu = E[f(X)]$. Let t denote real time and let $T(t)$ denote simulated time at real time t. We assume that simulated time grows linearly with real time, i.e., there exists a constant λ such that $T(t) = \lambda t$. Let $\hat{\mu}(t) = (1/T(t))\int_0^{T(t)} f(X_s)ds$ be the estimate of μ after a simulation of (real time) length t and let $\mu(t) = E[\hat{\mu}(t)]$ and $\sigma^2(t) = \mathrm{Var}[\hat{\mu}(t)]$. The bias of $\hat{\mu}(t)$ is $b(t) = \mu(t) - \mu$ and the mean squared error of $\hat{\mu}(t)$ is $\mathrm{mse}[\hat{\mu}(t)] = b^2(t) + \sigma^2(t)$.

Suppose that independent replications of $\mathbf{X}$ are run for t units of time on each of P processors producing iid estimates $\hat{\mu}_1(t), \ldots, \hat{\mu}_P(t)$ and an aggregate estimate $\bar{\mu}_1(t) = \sum_{p=1}^{P} \hat{\mu}_p(t)/P$. The mean squared error of $\bar{\mu}_1(t)$ is $\mathrm{mse}[\bar{\mu}_1(t)] = b^2(t) + \sigma^2(t)/P$.

Now assume that the P processors work together for t units of time on single realization of $\mathbf{X}$. Let $T_P(t)$ be the length, in simulated time, of this distributed simulation. We assume that there exists a constant α such that $T_P(t) = \alpha P T(t)$ where $T(t) = \lambda t$ is the simulated time achieved by a single processor in a run of real time length t. Thus the speed-up of the parallel simulation is αP and its efficiency is α. The distributed simulation estimate for μ is $\bar{\mu}_2(t) = (1/T_P(t))\int_0^{T_P(t)} f(X_s)ds$ and, since $T_P(t) = \alpha P T(t) = \alpha P \lambda t$, its mean squared error is $\mathrm{mse}[\bar{\mu}_2(t)] = b^2(\alpha P t) + \sigma^2(\alpha P t)$.

In order to compare these two methods specific assumptions need to be made concerning the bias $b(t)$ and the variance $\sigma^2(t)$. We assume that the bias is given by $b(t) = b/t$ for some constant b. This assumption can be shown to be true (to first order terms in $1/t$) for finite state space Markov chains (see chapter V of Doob (1953) for convergence rates to steady state), regenerative processes (Meketon and Heidelberger (1982)) or for processes for which there exists a time t_0 such that $E[X_s] = \mu$ for $s \geq t_0$. We shall assume that $\sigma^2(t) = \sigma^2/t$ which is true under very general conditions (Theorem 20.1 of Billingsley (1968) or Crane and Iglehart (1975)). Substituting $b(t) = b/t$ and $\sigma^2(t) = \sigma^2/t$ into the expressions for $\mathrm{mse}[\bar{\mu}_1(t)]$ and $\mathrm{mse}[\bar{\mu}_2(t)]$ yields the mean squared error ratio

$$r(P, \alpha, \gamma^2/t) = \frac{\mathrm{mse}[\bar{\mu}_2(t)]}{\mathrm{mse}[\bar{\mu}_1(t)]} = \frac{1/(\alpha P) + \gamma^2/(\alpha^2 P^2 t)}{1/P + \gamma^2/t}$$

where $\gamma = b/\sigma$. Thus the relative statistical efficiency is determined by three factors:

1. The number of processors P.
2. The speed-up factor α.
3. A term $\gamma^2/t = b^2/(\sigma^2 t)$ that takes into account the simulation run length, the strength of the initial transient and the variability of the process.

If $b = 0$, in which case there is no bias, then $r(P, \alpha, 0) = 1/\alpha \geq 1$ which means that replications is statistically more efficient. In this case there is no penalty in performing replications and the method has 100% efficiency. Similarly, replications is statistically more efficient for very long run lengths since $\lim_{t\to\infty} r(P, \alpha, \gamma^2/t) = 1/\alpha \geq 1$. On the other hand, if $b \neq 0$ then $\lim_{P\to\infty} r(P, \alpha, \gamma^2/t) = 0$, i.e., distributed simulation is statistically more efficient for a large number of processors provided that the efficiency α remains constant, or equivalently, the speed-up grows linearly with the number of processors. If the speed-up is given by an arbitrary function α_P such that $\lim_{P\to\infty} \alpha_P = \infty$, then $\lim_{P\to\infty} r(P, \alpha_P, \gamma^2/t) = 0$ which means that distributed simulation is asymptotically superior to replications as the number of processors increases. In this case, the replications estimate converges with probability one to $\mu + b/t$ which is the wrong answer.

Plots of the mean squared error ratio $r(P, \alpha, \gamma^2/t)$ are given in Figure 1 as a function of α for $P = 10$, $P = 20$ and several values of γ^2/t. Figure 1 shows that, for a given number of processors P and efficiency factor α, replications becomes sta-

tistically more efficient as γ^2/t decreases. The factor $\gamma^2/t = b^2/(\sigma^2 t)$ is small either because the run length t is large or because the bias term b is small relative to the variance term σ^2. Figure 1 also shows that, for a given efficiency α and γ^2/t, distributed simulation becomes more efficient as the number of processors P increases.

Although the M/M/1 queue is not particularly well suited for distributed simulation, it is interesting to compute its γ^2 since it could be indicative of the γ^2 of an open network of queues with corresponding traffic intensities. Blomqvist (1967) shows that $\sigma^2 = 4\rho/(1-\rho)^4 - \rho(2-\rho)/(1-\rho)^2$ for the mean waiting time in an M/M/1 queue with service rate $\mu = 1$ and traffic intensity ρ. For $\rho = 0.9$, $\sigma^2 = 35{,}901$ and a regression analysis of the M/M/1 simulation results presented in Table I of Meketon and Heidelberger (1982) yields the estimate $b = -1{,}269.6$ when the queue is started in the empty and idle state. Thus $\gamma^2 \simeq 44.9$. Note that these are the appropriate terms when t denotes the number of customers served. Table I of Heidelberger (1980) shows that approximately 120,000 customers must be simulated in order to achieve $\pm 10\%$ accuracy for the mean waiting time. Thus for a reasonably long simulation consisting of 10 processors and 12,000 customers/replication, $\gamma^2/t \simeq 0.0037$ which means that replications would provide a more accurate estimate unless the efficiency α is extremely close to one.

Now consider the effect of a simulation strategy that truncates the beginning portion of each replication in order to reduce the effects of the initial transient. Suppose there exists a t_0 such that $E[X_s] = \mu$ for $s \geq T(t_0)$ and that the portion between 0 and $T(t_0)$ is discarded. Since both estimates are now unbiased, the mean squared errors of the replications and distributed simulation estimates are $\sigma^2/(P(t-t_0))$ and $\sigma^2/(\alpha P t - t_0)$, respectively. Thus distributed simulation is statistically more efficient if and only if $\alpha \geq [(t-t_0)/t](P-1)/P$. For large values of P this condition is approximately $\alpha \geq (t-t_0)/t$ which means that distributed simulation is statistically more efficient than replications only when the efficiency α is greater than the fraction of the run that is in steady state. For example, if 10% truncation is required to eliminate the transient, the efficiency must be at least 0.90 for distributed simulation to produce more accurate estimates than replications.

3. COMBINING REPLICATIONS AND DISTRIBUTED SIMULATION

An alternative to using either replications or distributed simulation is to use a combination of the two methods. Suppose P processors are available and let R be the number of independent replications where each replication consists of a distributed simulation using M processors ($P = RM$). What is the optimal choice for R and M? The trade-off is between using distributed simulation to reduce the bias and using replications to increase the efficiency.

The mean squared error of the resulting estimate is $b^2(\alpha_M t) + \sigma^2(\alpha_M t)/R$. We will assume that the speed-up is given by $\alpha_M = M^\beta$ where $0 \leq \beta \leq 1$. The speed-up is thus assumed to be an increasing concave function of the number of processors.

Figure 2 plots the mean squared error as a function of M for $P = 32$ and several values of β and γ^2/t. These mean squared errors are normalized by the mean squared error obtained using iid replications ($M = 1$). Thus any value in Figure 2 that is greater than one means the method is less accurate than replications. This figure again shows that replications is statistically more efficient for either long runs, weak transients or low efficiencies. For the stronger transient ($\gamma^2/t = 0.10$), the optimal policy uses distributed simulation with only a small number of processors per replication except for very high efficiencies (β close to 1.0). Although not shown in this figure, if β and γ^2/t are fixed, then the optimal number of processors per replication increases as the number of processors increases.

4. STATISTICAL PROPERTIES OF PARALLEL INDEPENDENT REPLICATIONS

In this section we will discuss the statistical properties of parallel independent replications. Unlike the previous two sections which were concerned with steady state results, this section will deal with estimating a quantity from a transient, or terminating simulation.

The model is as follows. Assume there are P processors and that X_{ij} is the estimate from replication j on processor i. We shall be interested in estimating $\mu = E[X_{ij}]$. If a fixed number of replications, R_i, are run on processor i, then an unbiased estimate of μ is given by

$$\hat{\mu}(\mathbf{R}) = \frac{\sum_{i=1}^{P}\sum_{j=1}^{R_i} X_{ij}}{\sum_{i=1}^{P} R_i}.$$

Let $F(t)$ denote the distribution function of the time to complete a replication. If $F(t) < 1$ for all $t \geq 0$, then the expected completion time (the time until all processors finish) converges to infinity as $P \to \infty$.

Suppose now that a finite amount of computing time t is available on each processor for estimating μ via simulation. Thus replications are run on each processor for t units of real time

and a random number of replications are obtained. Let $N_i(t)$, $i = 1, \ldots, P$ be the number of replications completed on processor i by time t. One possible estimate for μ is

$$\hat{\mu}_1(t) = \frac{\sum_{i=1}^{P}\sum_{j=1}^{N_i(t)}\left(X_{ij}/N_i(t)\right)}{P}.$$

Under appropriate technical conditions, the results of Meketon and Heidelberger (1982) show that $E[\hat{\mu}_1(t)] = \mu + O(1/t)$ which means that $\hat{\mu}_1(t)$ will not converge to μ if t remains fixed as the number of processors increases. Another possible estimate of μ is

$$\hat{\mu}_2(t) = \frac{\sum_{i=1}^{P}\sum_{j=1}^{N_i(t)} X_{ij}}{\sum_{i=1}^{P} N_i(t)},$$

however, in a forthcoming paper Heidelberger (1986) shows that $E[\hat{\mu}_2(t)] = \mu + O(1/t) + O(1/tP)$ which again means that, for a fixed t, $\hat{\mu}_2(t)$ converges to the wrong quantity as the number of processors increases. Now consider the estimate

$$\hat{\mu}_3(t) = \frac{\sum_{i=1}^{P}\sum_{j=1}^{N_i(t)+1} X_{ij}}{\sum_{i=1}^{P}(N_i(t)+1)}$$

which requires letting each processor complete the replication in progress at time t. Heidelberger (1986) shows that $E[\hat{\mu}_3(t)] = \mu + O(1/tP)$. Thus $\hat{\mu}_3(t)$ converges to μ if either $t \to \infty$ or $P \to \infty$ whereas $\hat{\mu}_1(t)$ and $\hat{\mu}_2(t)$ converge to μ if and only if $t \to \infty$. The proofs rely on Wald's Equation ($E[\sum_{i=1}^{N} X_i] = E[N]E[X_i]$ if N is a stopping time and the X_i's are iid, see Chapter 4 of Karlin and Taylor (1975)) and the fact that $N_i(t) + 1$ is a stopping time whereas $N_i(t)$ is not a stopping time.

However, the price to be paid for using $\hat{\mu}_3(t)$ is an increased run length. As with $\hat{\mu}(\mathbf{R})$, if $F(s) < 1$ for all $s \geq 0$, then the expected completion time using $\hat{\mu}_3(t)$ converges to infinity as $P \to \infty$. For example, if the replication completion times are exponentially distributed ($F(t) = 1 - e^{-\lambda t}$), then the expected completion time is $t + (1/\lambda)\sum_{i=1}^{P}(1/i) \simeq t + \ln(P)/\lambda$. Using $\hat{\mu}_1(t)$ or $\hat{\mu}_2(t)$ with an equivalent expected completion time of $t' = t + (1/\lambda)\sum_{i=1}^{P}(1/i)$ would yield strongly consistent estimates for μ as $P \to \infty$. However, the bias of $\hat{\mu}_1(t)$ and $\hat{\mu}_2(t)$ would be of order $O(1/(t + \ln(P)))$ as opposed to the bias of $\hat{\mu}_3(t)$ which is of order $O(1/tP)$.

5. SUMMARY

This paper has considered the statistical properties of estimates obtained from discrete event simulations that are run on parallel processing computers. A model comparing the statistical accuracy of parallel distributed simulation and parallel independent replications was formulated. This model shows that, because of the random nature of simulations, a new set of factors must be considered in addition to the usual speed-up measure of a parallel computation's efficiency. These factors include the extent of the initial transient and the inherent variability of the system being simulated, the amount of computing time available and the number of processors. Generally speaking, if the run length is long or if the initial transient is weak, then replications will be statistically more efficient than distributed simulation in estimating steady state quantities. Given a reasonable speed-up factor, distributed simulation will be statistically more efficient than replications for short runs, for systems with a strong initial transient or if a large number of processors P are available and the speed-up $\alpha_P \to \infty$ as $P \to \infty$. This model was extended to consider a combination of replications and distributed simulation. If the number of processors is large and the transient is moderate relative to the run length, then the optimal policy uses distributed simulation with a small number of processors per replication and a large number of replications.

Finally, the problem of estimating a transient quantity using parallel independent replications was considered. The bias expansions of several different estimators were presented. If the expected time until all replications are completed is required to be finite, then none of these estimators are strongly consistent as the number of processors increases to infinity unless the computational time to complete a single replication is bounded.

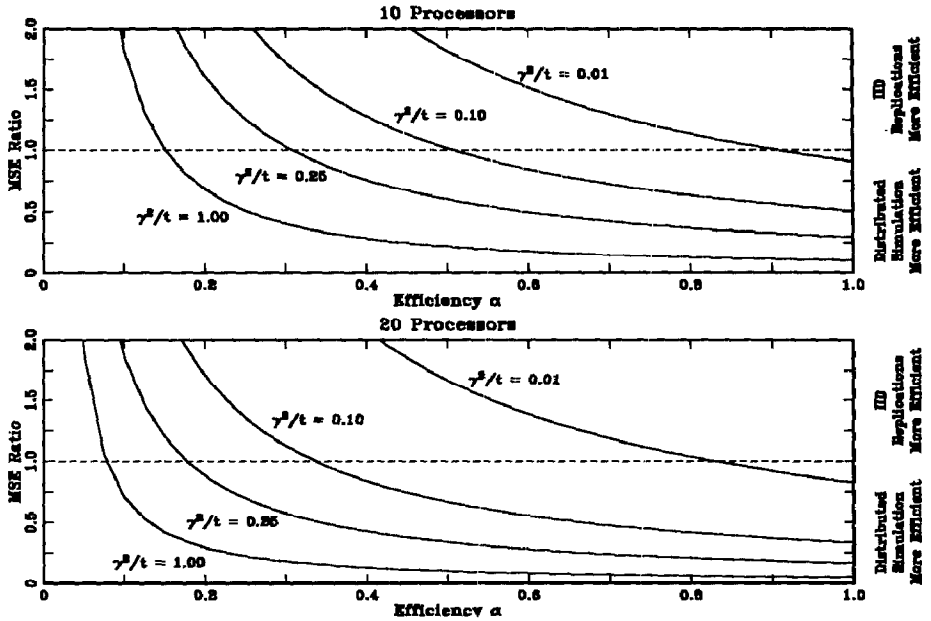

Figure 1: Comparison of the Mean Squared Errors of Replications and Distributed Simulation

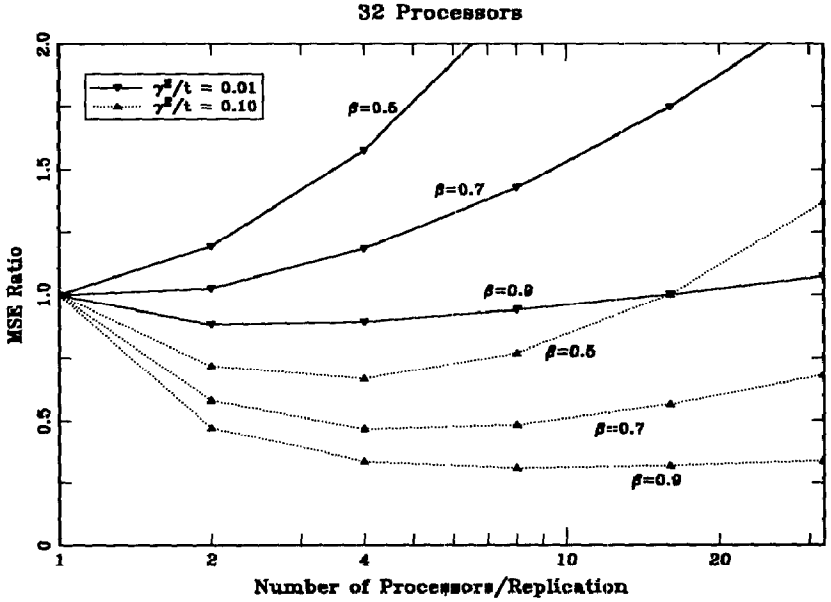

Figure 2: Relative Mean Squared Errors When Combining Replications and Distributed Simulation

REFERENCES

1. Berry, O. and Jefferson, D. (1985). Critical Path Analysis of Distributed Simulation. *Distributed Simulation 1985, The 1985 Society for Computer Simulation Multiconference*, San Diego, California.
2. Billingsley, P. (1968). *Convergence of Probability Measures*. John Wiley and Sons, Inc., New York.
3. Blomqvist, N. (1967). The Covariance Function of the M/G/1 Queue. *Skand. Akt. Tidskr.*, **50**, 157-174.
4. Chandy, K.M. and Misra, J. (1981). Asynchronous Distributed Simulation Via a Sequence of Parallel Computations. *Communications of the ACM* **24**, 198-206.
5. Comfort, J.C. (1983). The Design of a Multi-Microprocessor Based Simulation Computer - II. *Proceedings of the 16th Annual Simulation Symposium.* L.A. Holbrook (ed.), IEEE Computer Society Press, 197-209.
6. Comfort, J.C. (1984). The Simulation of a Master-Slave Event Set Processor. *Simulation* **42**, 117-124.
7. Crane, M.A. and Iglehart, D.L. (1975). Simulating Stable Stochastic Systems, III: Regenerative Processes and Discrete Event Simulations. *Operations Research* **23**, 33-45.
8. Doob, J.L. (1953). *Stochastic Processes*. John Wiley and Sons, Inc., New York.
9. Heidelberger, P. (1980). Variance Reduction Techniques for the Simulation of Markov Processes, I: Multiple Estimates. *IBM Journal of Research and Development* **24**, 570-581.
10. Heidelberger, P. (1986). Discrete Event Simulations and Parallel Processing: Statistical Properties. IBM Research Report (to appear), Yorktown Heights, New York.
11. Jefferson, D. and Sowizral, H. (1985). Fast Concurrent Simulation Using the Time Warp Mechanism. *Distributed Simulation 1985, The 1985 Society for Computer Simulation Multiconference*, San Diego, California.
12. Karlin, S. and Taylor, H.M. (1975). *A First Course in Stochastic Processes, Second Edition*. Academic Press, New York.
13. Lakshmi, M.S. (1979). A Study and Analysis of Performance of Distributed Simulation. Technical Report, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas.
14. Meketon, M.S. and Heidelberger, P. (1982). A Renewal Theoretic Approach to Bias Reduction in Regenerative Simulations. *Management Science* **28**, 173-181.
15. Wyatt, D.L., Sheppard, S. and Young, R.E. (1983). An Experiment in Microprocessor-Based Distributed Digital Simulation. *Proceedings of the 1983 Winter Simulation Conference, Volume 1*. S. Roberts, J. Banks and B. Schmeiser (eds.), IEEE, 271-277.

AUTHOR'S BIOGRAPHY

PHILIP HEIDELBERGER received a B.A. in mathematics from Oberlin College, Oberlin, Ohio, in 1974 and a Ph.D. in Operations Research from Stanford University, Stanford, California, in 1978. He has been a Research Staff Member at the IBM Thomas J. Watson Research Center in Yorktown Heights, New York since 1978. His research interests include computer performance modeling and statistical analysis of simulation output. He was manager of a performance modeling project from 1982 to 1985 and was the technical assistant to the director of IBM Research's Large Systems Laboratory from 1985 to 1986. Dr. Heidelberger is an Associate Editor of *Operations Research* and is a member of the Operations Research Society of America, the Association for Computing Machinery and the IEEE.

Philip Heidelberger
IBM Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598