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

RUN LENGTH CONTROL USING PARALLEL SPECTRAL METHOD

Kimmo E. E. Raatikainen

University of Helsinki, Department of Computer Science
Teollisuuskatu 23, SF-00510 Helsinki, FINLAND

ABSTRACT

Distributed and parallel simulation has been a popular research topic in recent years. The research has primarily concentrated on correctness and speedup of distributed simulation. The statistical output analysis that is an essential part of simulating stochastic systems has attained only small attention.

Parallel simulation is often mentioned as an attractive alternative for steady-state simulations. However, empirical results are not widely reported. In this study we present results of the parallel spectral method. The results are based on 12 simulation models executed in simulated parallel environment and on measured executing times and message passing delays.

1 INTRODUCTION

Since the fundamental contribution by Chandy and Misra (1981) a number of articles has been published covering various aspects of distributed and parallel simulation. The research has primarily concentrated on distributed simulation in which several processors cooperate on a single realization of the stochastic process simulated. The empirical studies — see e.g. Baik and Zeigler (1985), Comfort (1984), Duda (1989), Fujimoto (1988), Nicol (1988a and 1988b), Reed (1985), Reed and Malony (1988), Reed et al. (1988), Reynolds and Kuhn (1987), Wagner and Lazowska (1989) — have reported only modest speedups unless the queueing process simulated has a special structure.

Parallel simulation in which each processor simulates independent realizations of the stochastic process is an attractive alternative for steady-state simulations, see e.g. Rego and Sunderam (1992). The attraction arises from the fact that only minor modifications to uniprocessor simulation software are needed. The most notable drawback is that the size of the simulation model is restricted. Each processor must be able to simulate the whole model.

The research papers have considered primarily the correctness and speedup of the simulation. On the the other hand, statistical aspects of simulation have attained only minor attention. Notable exceptions include Heidelberger (1986 and 1988) and Glynn and Heidelberger (1990 and 1992). However, if we simulate a system having random input processes, the simulation study is only a programming exercise if the analysis of output processes is not properly carried out Pawlikowski (1990, p. 124).

In this paper we examine empirically one possible scheme for run length control in parallel simulation. The objective of run length control is to terminate the simulation as soon as the results are estimated to meet the given accuracy requirements. The importance of sequential estimation is widely recognized when stochastic queueing systems are examined through simulation.

The method examined combines the spectral method introduced in Heidelberger and Welch (1981) and the method of independent replications. Our objective is to find out whether a fixed number of independent replications executed in parallel and the spectral method can provide estimates that are accurate enough. Our second objective is to examine practical speedups gained through parallel execution. Finally, we want to learn the limitations of the method proposed.

In Section 2 we describe the method of parallel batch means. We give a brief mathematical description of the method. In addition, we outline the implementation. In Section 3 we report the observed coverages and estimated expected practical speedups. The results are based on 12 simulation models executed in simulated parallel environment and on measured executing times and message passing delays.

2 DESCRIPTION OF THE METHOD

2.1 Mathematical Description

The spectral method in uniprocessor simulation maintains non-overlapped batch means $\{\bar{X}_j\}_{j=1}^m$ of the output sequence $\{X_i\}_{i=1}^n$:

$$\bar{X}_j = \sum_{k=1}^{b} X_{(j-1)b+k}/b\,,\ n = mb\,.$$

The number of batches (m) varies from M to $2M-1$. When the number of batches with the current batch size (b) increases to $2M$, the current batch size is doubled and the number of batches is reduced to M by averaging two adjacent batches.

The accuracy of $\bar{X}(n) = \sum_{i=1}^{n} X_i/n$ as the estimate of $\mu = \mathrm{E}\{X\}$ is examined at prespecified checkpoints $n_1, n_2, \ldots, n_{max}$, $n_{i+1} = In_i$. The accuracy of the estimate is specified as the relative half-width of the confidence interval. The confidence interval estimated is based on assumption that the distribution of $(\bar{X}(n)-\mu)/s_{\bar{X}(n)}$ is the Student t-distribution with ν degrees of freedom. Therefore, the simulation is terminated at the first checkpoint ($n = \min\{n_1, \ldots, n_{max}\}$) in which

$$s_{\bar{X}(n)}/\bar{X}(n) \leq \varepsilon/t_\nu(1-\alpha/2)\,, \qquad (1)$$

where $s^2_{\bar{X}(n)}$ is the estimated variance of $\bar{X}(n)$ with ν degrees of freedom, ε is the relative accuracy of the confidence interval specified by the user, $1-\alpha$ is the confidence level specified by the user, and $t_\nu(x)$ is the $100x$th percentile of the Student t-distribution with ν degrees of freedom.

In the spectral method the estimated variance of $\bar{X}(n)$ is obtained through the periodogram of $\bar{X}_1, \ldots, \bar{X}_m$. A polynomial of order d is fitted to the first K ordinates of the bias corrected logarithms of the smoothed periodogram. Periodogram ordinates are evaluated using Fast Fourier transforms and smoothing is done by averaging two adjacent ordinates. For details, see Heidelberger and Welch (1981) and Pawlikowski (1990).

If we can generate multiple output sequences $\{X_{ki}\}_{i=1}^n$, $k = 1, \ldots, P$, in parallel, we have their batch means $\{\bar{X}_{kj}\}_{j=1}^m$, $k = 1, \ldots, P$, simultaneously available. The confidence interval for $\mu = \mathrm{E}\{X\}$ can be constructed using averages of batch means: $\bar{Y}_j = \sum_{k=1}^{P} \bar{X}_{kj}/P$ and $\hat{\mu} = \sum_{j=1}^{m} \bar{Y}_j/m$. The spectral method applied to sequence $\{\bar{Y}_j\}_{j=1}^m$ provides the estimated variance of $\hat{\mu}$. As in the uniprocessor case, the simulation is terminated at the first checkpoint in which

$$s_{\hat{\mu}}/\hat{\mu} \leq \varepsilon/t_\nu(1-\alpha/2)\,. \qquad (2)$$

Mathematically the termination rules (1) and (2) are asymptotically ($\varepsilon \to 0$) equivalent, when P is fixed and $n_{i+1} = n_i + 1$. In practice, the rules may have different properties when $\varepsilon > 0$. The frequency of checkpoints ($n_i = I^{i-1}n_1$) also affects the properties.

2.2 Outline of Implementation

When we can simultaneously execute several processes, the parallel simulation can be arranged as follows. One process is the master and the rest are slaves. Each slave receives simulation request messages from the master. Two kind of requests are needed:

`initialize`: The slave obtains initial seed, batch size (b), and the number of batches (m). It initializes the model, generates the first bm observations, and sends m batch means in the reply message.

`simulate`: The slave obtains batch size (b) and the number of bathes (m). It generates the next bm observations, and sends m batch means in the reply message.

The master initializes the parallel simulation by issuing the `initialize` request to each slave. When the master obtains a reply message, it identifies the slave sending the message, recognizes the checkpoint (n_i) the slave reached, calculates the batch size and number of batches needed to reach the next checkpoint (n_{i+1}), and sends the `simulate` request to the slave. When each slave have reached the checkpoint (n_i), the master generates the confidence interval using the sequence $\bar{Y}_1, \ldots, \bar{Y}_m$. If the interval is narrow enough, the master sends the KILL-signal to each slave.

3 EMPIRICAL RESULTS

We examined the properties of the parallel spectral method outlined in the previous section using 12 different simulation models and 6 different accuracy requirements. As stated in the Introduction our objective was to get an answer to the following questions:

1. Does the parallel spectral method provide estimates that meet the accuracy requirement prespecified by the user?
2. Is it possible, in practice, to obtain remarkable speedup through parallel generation of observations?

3. What kind of limitations for the method does exist?

The answers to these questions are based on results obtained through simulated parallelism. In addition, we have used measured execution times and message passing delays in a local area network. The decision to use simulated parallelism and measurements was based on the fact that it was our only possibility to obtain completely controllable experimental environment. The simulations were executed in a VAX8800. The execution times were measured in a dedicated SPARCstation IPC. The message passing delays were measured using two SPARCstation IPCs connected by 10 Mbs Ethernet. Before presenting the results we describe the experiments.

3.1 Description of Experiments

The simulation models examined are taken from the survey of sequential procedures for steady-state simulation by Law and Kelton (1982) and from the paper introducing the spectral method by Heidelberger and Welch (1981).

Models 1–9 are exactly the same as in Law and Kelton (1982). Model 10 is a slight modification of Model 10 in Law and Kelton (1982). Models 11 and 12 are the queueing network models used in Heidelberger and Welch (1981). Appendix A gives detailed descriptions of the models.

The internal parameters of the spectral method were: $M = 512$, $d = 2$, $K = 31$, $n_1 = 512$, $n_{max} = 8\,388\,608$, $I = 2$, and $\nu = 9$.

In the experiments we examined six pairs of (ε, α) specifying the accuracy requirement: $\varepsilon = 0.10$ and 0.05; $\alpha = 0.10$, 0.05, and 0.01. The number of slave processes examined was $1, 2, \ldots, 20$. An experiment consisted of two phases. In the first phase we generated 20 independent sequences of observations. The sequence length was n_{max}. At each checkpoint we recorded the batch means $\{\bar{X}_{kj}\}_{j=1}^{m}$, $k = 1, \ldots, 20$. In the second phase we used these batch means to determine the termination point and to generate the confidence interval. When the number of slave processes examined was P, replications $k = 1, \ldots, P$ were used.

With each model 100 independent experiments were executed. The simulations were initialized using the known (Models 1–3, 5–8, 10–12) or approximated (Model 4) steady-state distribution for the number of customers at different service centers. Model 9 was initialized as in Law and Kelton (1982).

Table 1: Number of models for which the method was valid: *The method is valid if the hypothesis that the true coverage is at least $1 - \alpha$ cannot be rejected at significance level α*

Number of slave processes	ε = 0.10			ε = 0.05		
	α = 0.10	α = 0.05	α = 0.01	α = 0.10	α = 0.05	α = 0.01
1	10	11	12	12	12	12
2	10	9	12	12	12	12
3	10	12	12	10	12	12
4	12	12	12	12	12	12
5	11	12	12	12	12	12
6	11	12	12	11	12	12
7	11	11	12	12	12	12
8	10	11	12	12	12	12
9	11	11	12	12	12	11
10	11	11	12	11	12	11
11	11	11	12	11	12	12
12	11	12	12	11	12	12
13	10	12	12	12	12	12
14	12	12	12	12	12	12
15	12	12	12	12	11	12
16	9	12	12	11	11	12
17	9	12	12	12	12	12
18	8	12	12	11	12	12
19	9	11	12	12	12	12
20	11	11	12	12	12	12

3.2 Observed Coverage

The question whether the estimates satisfy the prespecified accuracy requirement is examined through analyzing the observed coverages. The observed coverage is the fraction of experiments in which the estimated confidence interval, $[\hat{\mu} - s_{\hat{\mu}} t_v(1 - \alpha/2), \hat{\mu} + s_{\hat{\mu}} t_v(1 - \alpha/2)]$, includes μ. The number of experimental points ($12 \times 20 \times 6 = 1440$) is too large to allow the presentations of all the observed coverages. Therefore, we only give a summary.

Following Lavenberg and Sauer (1977) we regard the method as valid for a particular model and accuracy requirement (ε, α) if the observed coverage does not provide evidence enough to reject at significance level α the hypothesis that the true coverage is at least $1 - \alpha$. Since the experiments were independent, the observed coverage has binomial distribution with parameters n and p, where n is the number of experiments and p is the true coverage. Table 1 gives the number of models for which the method is valid.

Table 2: Measured Execution Times

T_{setup}	2.61 ms	Setup time of Cholesky fractorization and Fast Fourier Transform used in spectral method (mean of 1 000 replications)
T_{delay}	5.67 ms	Delay due to sending request and receiving reply using Internet socket (mean of 3 000 sends and receives)
$T_{non-overlapped}$	1.39 ms	Non-overlapped part (in master) of T_{delay}
T_{anal}	38.17 ms	Time needed to estimate the confidence interval using the spectral method (mean of 1 000 replications)
$T_{update}(p)$, $p > 1$	$1.17 + 0.58p$ ms	Time needed in maintaing batch means (mean of 1 000 replications and lsq-fit)
$T_{update}(1)$	0.26 ms	(mean of 1 000 replications)
$T_{gene}(n)$	*	Time needed (in slave) to generate n observations (measured execution times in 100 replications for $n_1, n_2, \ldots$)
$T_{(k\|p)}(n)$	*	Expected time needed to obtain the k^{th} set of n observations, when p sets are generated in parallel (continuous, piecewise-linear empirical distribution function from measured $T_{gene}(n)$'s)
$\Pr\{n_j\|p\}$	*	Fraction of experiments in which the simulation with p slave processes was terminated at the checkpoint n_j

*: Highly model depended; Not reported in this papers but are available from the author.

The overall conclusion is that the parallel spectral method usually provides valid estimates for the mean. However, when the number of slave processes is high and the accuracy requirement is not stringent, it is possible that observed coverage is low. In the light of results reported for the method of independent replications this is not surprising. If the number of slave processes is high and the accuracy requirement does not force the simulation to be continued until the Student t-approximation applied in the spectral method is accurate enough, the constructed confidence intervals will usually have a low coverage.

3.3 Expected Practical Speedups

The results above were based on simulated parallelism. Now we give estimated practical speedups. These results are based on observed sequence lengths and on measured execution times in two SPARCstation IPCs connected through 10 Mbs Ethernet.

We assume that 1 processor is executing the master processes and p processors are dedicated for p slave processes. The expected practical speedups with $p+1$ processors, $S(p+1)$, are based on estimated means of turnaround times:

$$S(p+1) = T_u/T_p \, , \; p \geq 1 \, .$$

The estimated mean in uniprocessor simulation, T_u, is:

$$T_u = T_{setup} + \sum_{j=1}^{max} \Pr\{n_j|1\}\left(T_{(1|1)}(n_j) + jT_{anal}\right) . \quad (3)$$

The estimated mean in parallel simulation with $p+1$ processors, T_p, is:

$$\begin{aligned} T_p \;=\; & T_{setup} + T_{delay} + T_{update}(p) + T_{anal} + \\ & \sum_{j=1}^{max} \Pr\{n_j|p\}(T_{(p|p)}(n_j) + D_p(n_j)) . \end{aligned} \quad (4)$$

The terms in the expressions above are explained in Table 2 that also contains their measured values. In (4) the term $T_{delay} + T_{update}(p) + T_{anal}$ is the time spent in the last construction of the confidence interval. The term $D_p(n_j)$ is the delay in the master if the previous constructions of confidence intervals and the updates of batch means do not overlap with generation of observations:

$$D_p(n_j) = \begin{cases} 0 \, , & j = 1 \, , \\ D_p(n_{j-1}) + G_p(n_j) \, , & j > 1 \, , \end{cases} \quad (5)$$

where

$$\begin{aligned} G_p(n_j) \;=\; & \max\Big\{0, T_{delay} + T_{update}(p) + T_{anal} \\ & -\left(T_{(1|p)}(n_j) - T_{(p|p)}(n_{j-1})\right)\Big\} \\ & + \sum_{k=2}^{p} \max\Big\{0, T_{non-overlapped} + T_{update}(p) \end{aligned}$$

$$-\left(T_{(k|p)}(n_j) - T_{(k-1|p)}(n_j)\right)\Big\} .$$

Since the delay $D_p(n_j)$ is cumulative, it corresponds to a frequent synchronization of the master and the slaves. The first term of $G_p(n_j)$ is the possible delay due to the fact that the construction of the confidence interval at the previous checkpoint is not ready, when the first slave reaches the current checkpoint. The term in the sum are the possible delays due to the fact that the update of the batch means cannot be done between the arrivals of replies.

Figures 1a and 1b show the expected practical speedups separately for each model. The figures indicate that the expected speedup heavily depends on model. The overall conclusion is that we usually obtain moderate speedups only when the number of slave processes is less than ten. The primary reason that restricts the speedup is the variance of execution times in generating the observations. When the number of processor is high, the variation significantly reduces the speedup, since each slave must reach the checkpoint before the confidence interval can be constructed.

In Model 4 the overlinear speedup is due to the fact that the sequence length grows exponentially, i.e. $n_i = I^{i-1} n_1$. However, the coverage is not valid in the most of the experimental points showing overlinear speedup. The explanation for Model 9 is that the simulation is so simple that the run is always terminated at $n_1 = 512$.

3.4 Limitations of the Method

The results reported above indicate that the method should only be used when the number of slave processes is moderate, say from 5 to 10. If the number of slave processes is high, we have

- a high possibility to obtain estimates that does not satisfy the accuracy requirement specified by the user,
- a high possibility to waste processing time, since the same or better speedup can be obtained with fewer processes.

4 SUMMARY

We have introduced a parallel spectral method and outlined one possible implementation. Results from extensive simulation experiments indicate that the method is an attractive way of using parallelism in networks of 5–10 workstations.

The observed coverages indicate that method provides valid confidence intervals, when the number of slave processes is not very high. The estimated turnaround times indicate that the variation of the execution times needed in generating the observations significantly reduces the speedup, when the number of processors increases.

Before we implement a parallel spectral method, we want to develop a method for detecting the initialization bias. In addition, we want to examine various communication and synchronization schemes. Particularly, we are interested in schemes that allow the confidence interval to be constructed without waiting for each slave process to reach the checkpoint. Such a scheme improves both the efficiency and reliability of the method.

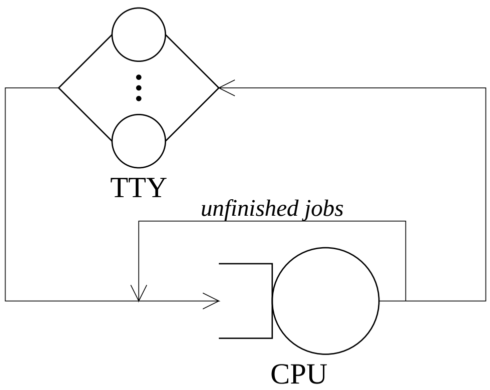
APPENDIX A: DESCRIPTION OF MODELS EXAMINED

Model 1: Mean waiting time in M/M/1-queue: $\lambda = 1$, $\mu = 1.25$, FCFS

Model 2: Mean waiting time in M/M/1-queue: $\lambda = 1$, $\mu = 1.25$, LCFS

Model 3: Mean waiting time in M/M/1/M/1-tandem queue: $\lambda = 1$, $\mu_1 = \mu_2 = 1.25$, FCFS

Model 4: Mean turnaround time in Time-Shared Computer Model:



Population 35
$\mu_{\text{TTY}} = 0.04$
$\mu_{\text{CPU}} = 1.25$
round robin scheduling at CPU
– quantum 0.1
– switch overhead 0.015

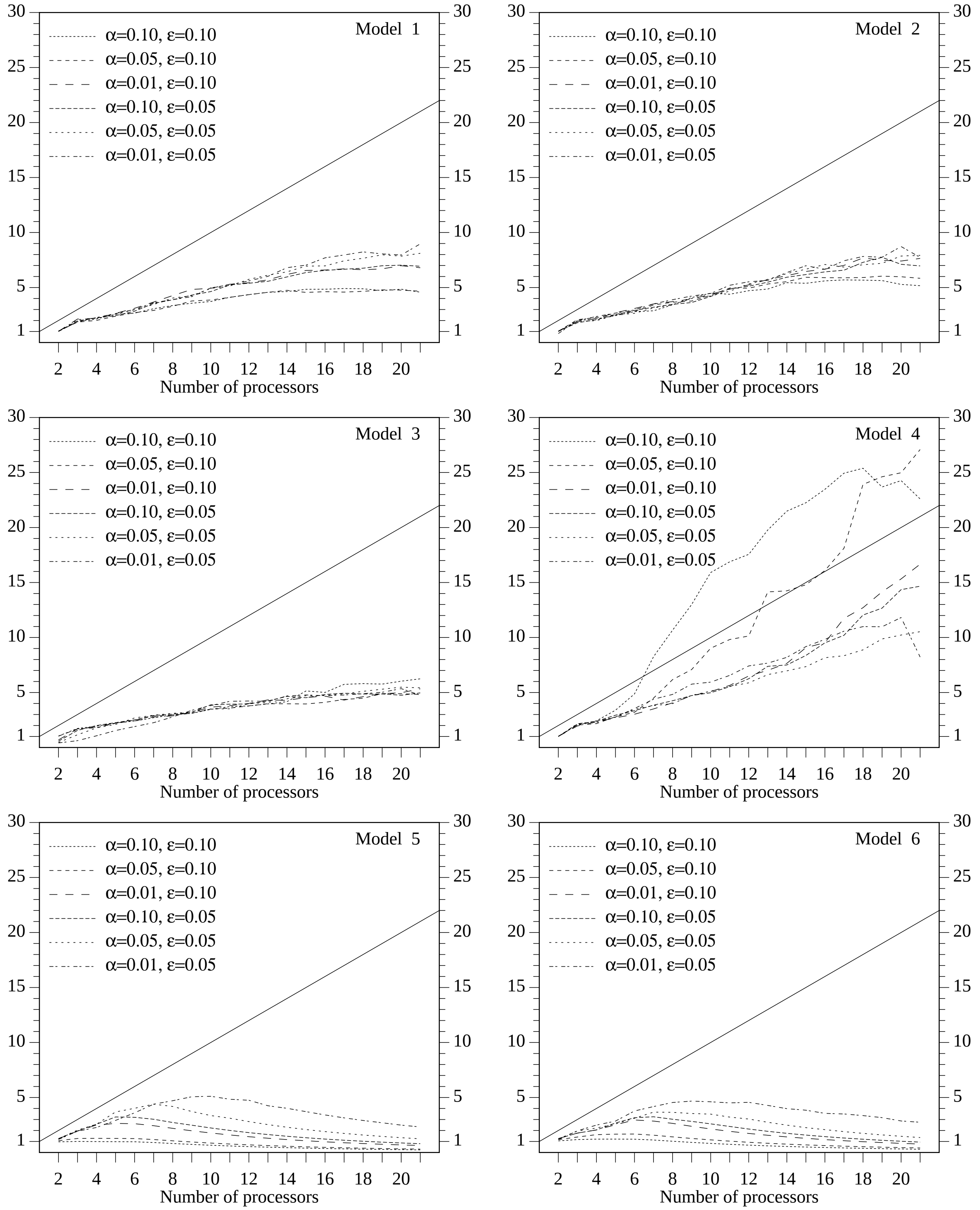


Figure 1a: Expected practical speedup

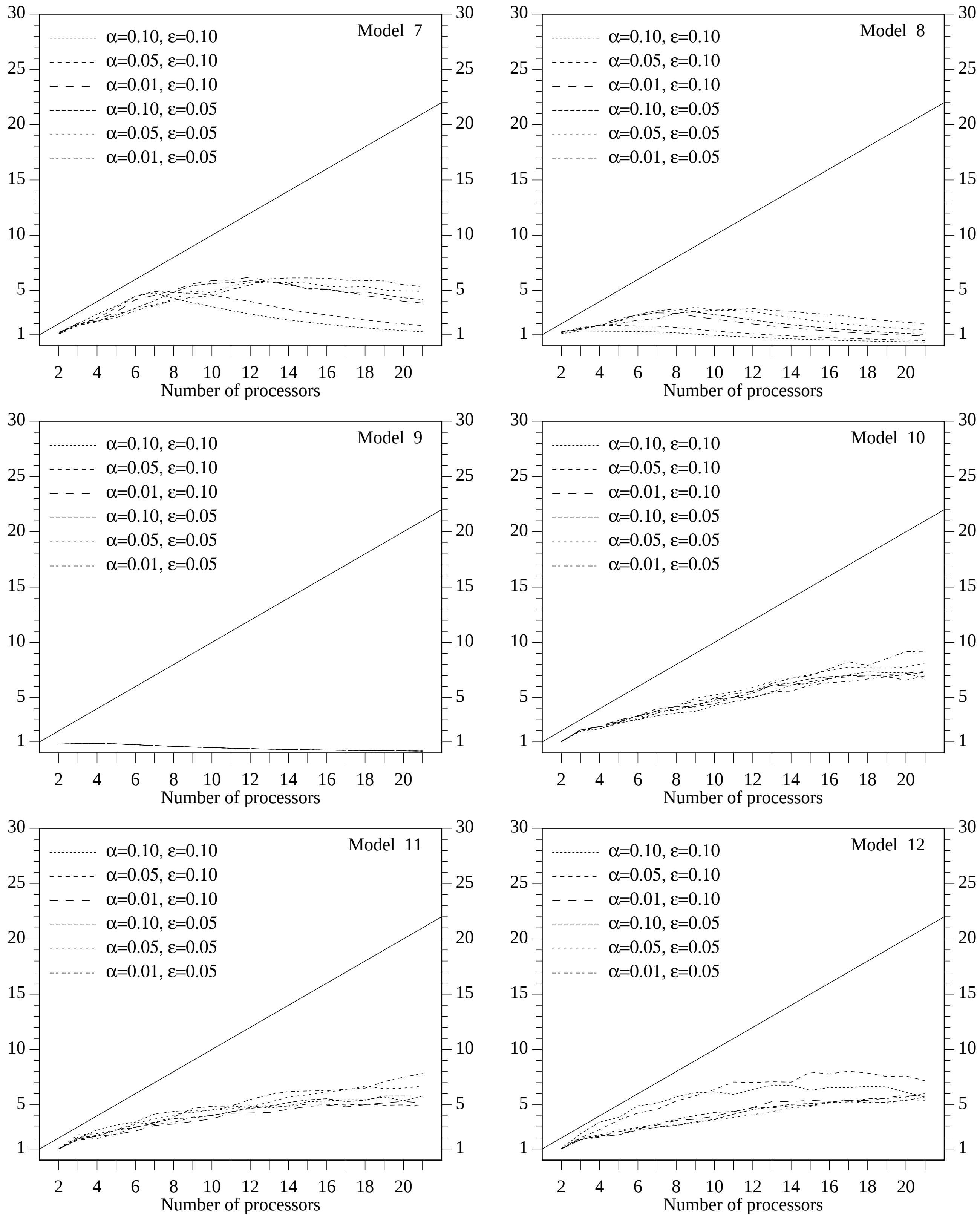
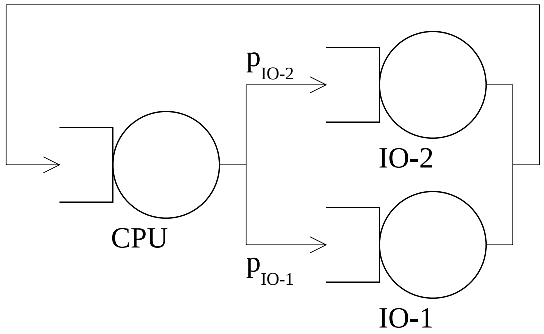


Figure 1b: Expected practical speedup

Model 5–8: Mean round trip time in Central-Server Computer Model:



Model	5	6	7	8
Population	4	8	8	8
μ_{CPU}	1	1	1	1
$\mu_{\text{IO-1}}$	0.5	0.5	0.45	1.8
$\mu_{\text{IO-2}}$	0.5	0.5	0.05	0.2
$p_{\text{IO-1}}$	0.5	0.5	0.9	0.9
$p_{\text{IO-2}}$	0.5	0.5	0.1	0.1

Model 9: Expected total cost during one period in (s,S) Inventory System:

Let X_i be the amount of inventory on hand before ordering, Y_i the amount of inventory on hand after ordering, and Q_i the demand, each in period i.

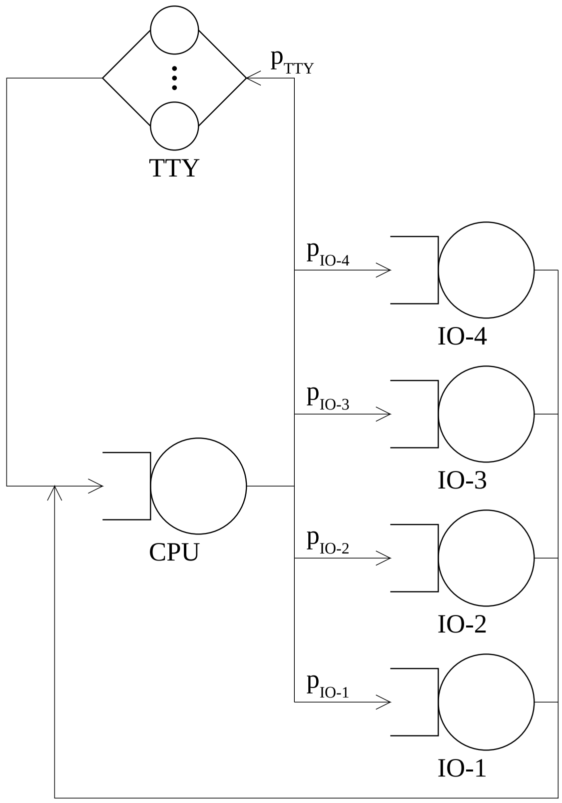
If $X_i < s$, then $S - X_i$ items are ordered ($Y_i = S$) and an ordering cost $K + c(S - X_i)$ is incurred. If $X_i \leq S$, then no order is placed ($Y_i = X_i$) and no ordering cost is incurred.

After Y_i has been determined, then a demand Q_i occurs. If $Y_i \geq Q_i$, then a holding cost $h(Y_i - Q_i)$ is incurred. If $Q_i > Y_i$, then a shortage cost $\pi(Q_i - Y_i)$ is incurred. In either case, $X_{i+1} = Y_i - Q_i$.

Parameters: $s = 17$, $S = 57$, $K = 32$, $c = 3$, $h = 1$, $\pi = 5$, $X_1 = S$. The demands (Q_i's) are independent Poisson random variables with mean 25.

Model 10: Mean queue length at arrival instance in M/M/1-queue: $\lambda = 1$, $\mu = 1.25$, FCFS

Model 11–12: Mean response time in Central-Server Computer Model:



Model	11	12
Population	25	25
μ_{TTY}	0.01	0.01
μ_{CPU}	1.0	1.0
$\mu_{\text{IO-1}}$, $\mu_{\text{IO-2}}$	0.72	0.18
$\mu_{\text{IO-3}}$, $\mu_{\text{IO-4}}$	0.08	0.04
$p_{\text{IO-1}}$, $p_{\text{IO-2}}$	0.36	0.36
$p_{\text{IO-3}}$, $p_{\text{IO-4}}$	0.04	0.04
p_{TTY}	0.2	0.2

REFERENCES

Baik, D., and B. P. Zeigler. 1985. Performance evaluation of hierarchical distributed simulators. In *Proceedings of the 1985 Winter Simulation Conference*, ed. D. D. Gantz, G. C. Blais, and S. L. Solomon, 421–427. Institute of Electrical and Electronics Engineers.

Chandy, K. M., and J. Misra. 1981. Asynchronous distributed simulation via a sequence of parallel simulations. *Communications of the ACM* 24:198–206.

Comfort, J. C. 1984. The simulation of a master-slave event set processor. *Simulation* 42:117–124.

Duda, A. 1989. On the tradeoff between parallelism and communication. In *Modeling Techniques and Tools for Computer Performance Evaluation*, ed. R. Puigjaner and D. Potier, 323–334. New York: Plenum Press.

Fujimoto, R. M. 1988. Performance measurements of distributed simulation strategies. In *Proceedings of the SCS Multiconference on Distributed Simulation, 1988*, ed. B. Unger and D. Jefferson, 14–20: Society for Computer Simulation.

Glynn, P. W., and P. Heidelberger. 1990. Bias properties of budget constrained simulations. *Operations Research* 28:801–814.

Glynn, P. W., and P. Heidelberger. 1992. Experiments with initial transient deletion for parallel, replicated steady-state simulations. *Management Science* 38:400–418.

Heidelberger, P. 1986. Statistical analysis of parallel simulation. In *Proceedings of the 1986 Winter Simulation Conference*, ed. J. R. Wilson, J. O. Henriksen, and S. D. Roberts, 290–295. Institute of Electrical and Electronics Engineers.

Heidelberger, P. 1988. Discrete event simulation and parallel processing: statistical properties. *SIAM Journal of Statistics and Computation* 9:1114–1132.

Heidelberger, P., and P. D. Welch. 1981. A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM* 24: 233–245.

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

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

Nicol, D. M. 1988a. Parallel discrete event simulation of FCFS stochastic queueing networks. In *Parallel Programming: Experience with Applications, Languages and Systems.* 124–137. ACM SIGPLAN.

Nicol, D. M. 1988b. High performance parallelized discrete-event simulation of stochastic queueing networks. In *Proceedings of the 1988 Winter Simulation Conference*, ed. M. A. Abrams, D. L. Haigh, and J. C. Comfort, 306–314. Institute of Electrical and Electronics Engineers.

Pawlikowski, K. 1990. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys* 22:123–170.

Reed, D. A. 1985. Parallel discrete event simulation: a case study. In *Proceedings of the 18th Annual Simulation Symposium*, ed. A. Miller, 95–107. IEEE Computer Society Press.

Reed, D. A., and A. D. Malony. 1988. Parallel discrete event simulation: the Chandy-Misra approach. In *Proceedings of the SCS Multiconference on Distributed Simulation, 1988*, ed. B. Unger and D. Jefferson, 8–13. Society for Computer Simulation.

Reed, D. A., A. D. Malony, and B. D. McCredie. 1988. Parallel discrete event simulation using shared memory. *IEEE Transactions on Software Engineering* SE-14: 541–553.

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

Reynolds, P. F., and C. S. Kuhn. 1987. A performance study of three protocols for distributed simulation. In *Proceedings of the Conference on Methodology and Validation, 1987*, ed. O. Balci, 26–31. Society for Computer Simulation.

Wagner, D. B., and E. D. Lazowska. 1989. Parallel simulation of queueing networks: limitations and potentials. In *Proceedings of the 1989 ACM SIGMETRICS and Performance '89*, 146–155. ACM SIGMETRICS.

AUTHOR BIOGRAPHY

KIMMO E. E. RAATIKAINEN is an Assistant Professor in the Department of Computer Science at University of Helsinki, Finland. He received M.S. and Ph.D. degrees in computer science from University of Helsinki in 1983 and 1990 respectively. His research interests are focused on statistical aspects of queueing network simulation and on evaluation of distributed systems and applications.