Speeding Up Stochastic Discrete-Event Simulation

Krzysztof Pawlikowski and Donald McNickle

University of Canterbury
Christchurch, New Zealand
E-mail: K. Pawlikowski@cosc.canterbury.ac.nz

Keywords: sequential stochastic simulation, simulation speedup, distributed and parallel simulation.

Abstract

Stochastic discrete-event simulation with sequential analysis of output data at consecutive checkpoints allows full control of the errors of the simulation results. Unfortunately, applications of sequential stochastic simulation are impeded by the fact that very long simulation runs can be required to reduce statistical errors of results to an acceptable level. In this paper we formulate rules governing the speedup of sequential stochastic simulation executed under the Multiple Replications in Parallel scenario. In this scenario, output data are produced in parallel by multiple processors that run independent replications of a given simulation and submit the generated data to global analysers. Both theoretical and practical implications of the rules governing speedup are discussed.

1 Introduction

Stochastic discrete-event simulation has become the most popular tool of performance evaluation of such complex stochastic dynamic systems as, for example, telecommunication networks. For drawing conclusions from such performance evaluation studies, one needs to be sure that statistical errors associated with simulation results are appropriately small, since it is well documented that the conclusions on the basis of results with high (or unknown) statistical errors can be very misleading; see for example (Pawlikowski 1999). Because of that, simulation conducted for a system's performance evaluation should be run sequentially, i.e. it should follow a sequence of consecutive checkpoints at which the accuracy of estimates is assessed, until the statistical errors of results at some checkpoint falls below an acceptable threshold and the simulation can stop. **Sequential stochastic simulation** is the only practical approach allowing control of the error of the final results, since *"... no procedure in which the run length is fixed before the simulation begins can be relied upon to produce a confidence interval that covers the true <value> with the desired probability level"* (Law and Kelton 1991).

Unfortunately, applications of sequential stochastic simulation are impeded by the fact that very long simulation runs can be required to reduce statistical errors of results to an acceptable level. This is especially true when the output data come from strongly correlated time series, as is the case in telecommunication networks with teletraffic modelled by stochastically self-similar and long range dependent processes. One traditional approach to solve this problem would be to reduce the simulation run length by reducing the variances of estimators used in the simulation, i.e. by applying one of the Variance Reduction Techniques (VRTs) (Law and Kelton 1991).

Another possible way of reducing the simulation run length is to reduce the time needed for collecting the required sample of output data by generating these data in parallel, on multiple processors. This means launching multiple independent replications of the same simulation on multiple simulation engines for concurrent production of statistically equivalent output data. Streams of output data coming from different simulation engines are delivered to a global analyser, responsible for calculating estimates and their errors at consecutive checkpoints. This scenario of sequential stochastic simulation is known as **Multiple Replications in Parallel** or **MRIP** (Pawlikowski et al. 1994), and was first suggested in (Bhavsar and Isaac 1987). Since then a few implementations of this idea have been reported. These include AKAROA.2, a controller of stochastic simulation designed at the University of Canterbury in Christchurch, New Zealand (Ewing et al. 1999).

In this paper we formulate rules governing the speedup of sequential stochastic simulation executed under the MRIP scenario, both for terminating and steady-state stochastic simulations. The application of these rules in practical simulation studies will also be discussed. Without losing the generality of our findings, we will restrict our discussion to the MRIP simulations being executed on multiple processors of a local area network.

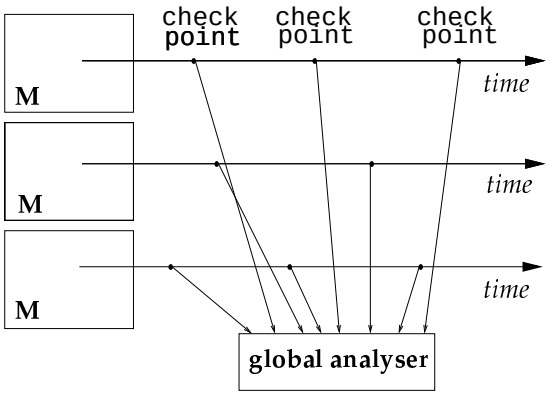

Figure 1: Distributed stochastic simulation in MRIP scenario: three simulation engines, each executing simulation of the model M.

2 A generic implementation of MRIP and its speedup

Execution of sequential stochastic simulation under the Multiple Replications in Parallel scenario means launching multiple independent replications of a simulation on multiple processors or workstations of a local network. These workstations operate as simulation engines that concurrently produce mutually independent streams of (statistically equivalent) output data, or observations, to be delivered for analysis to a global analyser, see Figure 1.

Each simulation engine follows its own sequence of checkpoints, submitting its portion of the pooled sample of observations to the global analyser whenever it reaches its next checkpoint. Let us note that, since observations used by such analysers come from different simulation engines, some performance measures or methods of analysis may need to use specially defined estimators, that take into account the distributed origin of analysed output data; see for example (Pawlikowski et al. 1994), for an MRIP estimator of a steady-state mean, that is a generalization of a sequential non-MRIP estimator, based on Spectral Analysis (Heidelberger and Welch 1981).

Different policies of synchronisation or workload balancing of multiple simulation engines have been proposed and studied in (Bhavsar and Isaac 1987), (Heidelberger 1988) and (Lin 1994). However, based on our experience during almost 10 years of research on distributed stochastic simulation with AKAROA project (see www.cosc.canterbury.ac.nz/research/RG/net_sim/simulation_group/akaroa/), we assume that *simulation engines participating in stochastic simulation in the MRIP scenario do not need any supervision or synchronization*, since the best solution is to give each of them a chance to contribute in the collective production of observations to such an extent as each of them can or is able to. This strategy of **uncoordinated cooperation** allows finishing simulation in the shortest possible time within a given computing environment, while minimizing communication overhead.

Of course, we should remember that any sequential stochastic discrete-event simulation in the MRIP scenario represents a simulated sequential statistical experiment. Thus, extra precautions have to be taken against the possibility of accidental stopping of sequential simulation having collected too few observations, when the statistical error becomes only temporarily acceptable. In general, too short samples of output data can give biased results with poor coverage. To avoid such too-short simulation runs, one should ensure appropriate location of the first checkpoint. Additionally, one can use special heuristic rules protecting against too-short simulation runs (Lee et al. 1999). Once these precautions are taken, one can expect not only unbiasedness of the final results from sequential stochastic simulation executed in the MRIP scenario, but also a speedup due to the distributed nature of output data generation.

Employing multiple simulation engines for concurrent production of output data, one can expect that the speedup achievable under the MRIP scenario is governed by a version of Amdahl's law, formulated for assessing reduction of the processing time of a task if it is processed concurrently by a number of processors, (Amdahl 1967).

To determine such a law for sequential stochastic simulation in the MRIP scenario on P simulation engines, let us measure the processing task associated with a given simulation by the total number of observations that are needed for achieving a declared level of statistical errors of the results. Additionally, let us temporarily assume that each of P simulation engines makes exactly the same contribution to a given simulation, and that checkpoints followed by each simulation engine are located exactly D observations apart.

Let N_{min} observations be required for obtaining sufficiently accurate final results when a single simulation engine is used, N_P observations be generated by each of P simulation engines, and f $(0 \leq f < 1)$ be the fraction of the sample of N_{min} observations that cannot be collectively produced by multiple simulation engines. Additionally, for simplification,let us temporarily assume that $(1-f)N_{min} = PN_P$.

Then the resulting speedup $S_P = N_{min}/N_P)$ is given by the following formula:

$$S = \begin{cases} \frac{1}{f+(1-f)/P}, & \text{for } P \leq \frac{(1-f)N_{min}}{D}; \\ \frac{N_{min}}{fN_{min}+D}, & \text{for } P \geq \frac{(1-f)N_{min}}{D}. \end{cases} \quad (1)$$

This formula states that the maximum speedup equals

$$S_{max} = \frac{N_{min}}{fN_{min}+D},$$

and is achieved when one uses $P_{max} = \frac{(1-f)N_{min}}{D}$ simulation engines, i.e. all required observations are available when the simulation engines reach their first checkpoint. Note that the speedup cannot be increased by introducing more than P_{max} simulation engines: such an action would only decrease the final error of the results, making them more accurate than required.

Having derived Equation (1), we note that, by the virtue of stochastic simulation, each replication will have different run length, i.e. N_{min} assumes different values in different replications. Thus, we should introduce $\bar{N}_{min}$, the mean number observations needed for stopping simulation when a single simulation engine is used. In the case of P simulation engines, each of them should use statistically independent streams of pseudorandom numbers. Thus, we should introduce: $\bar{f}$ $(0 \leq \bar{f} < 1)$ as the mean fraction of the $\bar{N}_{min}$ observations that cannot be produced in parallel[1], and $\bar{D}$ as the mean distance between checkpoints (the number of observations between consecutive checkpoints, averaged over all simulation engines). Then, for

$$N_P = \lceil \frac{(1-\bar{f})\bar{N}_{min}}{P} \rceil,$$

where $\lceil x \rceil$ denotes the largest integer that is smaller than $x+1$, the formula governing the mean speedup of sequential stochastic simulation in the MRIP scenario is given as follows:

$$\bar{S} = \begin{cases} \frac{1}{\bar{f}+(1-\bar{f})/P}, & \text{for } P \leq \lceil \frac{(1-\bar{f})\bar{N}_{min}}{\bar{D}} \rceil; \\ \frac{\bar{N}_{min}}{\bar{f}\bar{N}_{min}+\bar{D}}, & \text{for } P \geq \lceil \frac{(1-\bar{f})\bar{N}_{min}}{\bar{D}} \rceil. \end{cases} \quad (2)$$

The dependence of $\bar{S}$ on $\bar{f}$ and P is depicted in Figure 2, for $\bar{N}_{min} = 10000$ and $D = 1$. One can see that the speedup is a linear function of the number of simulation engines used only if all observations can be cooperatively produced in parallel ($\bar{f} = 0$), and $\bar{S}_{max}$ strongly degrades as $\bar{f}$ increases.

Since we have assumed a homogeneous set of processors operating as simulation engines, whose cooperative effort is left totally unsupervised, Equation (2) determines **the upper bound of mean speedup** of sequential stochastic simulation in the MRIP scenario, executed in a homogeneous computing environment. In the extremely asymmetric environment, if one simulation engine uses very fast processor and the remaining processors are very slow, the MRIP scenario can offer no gain in speedup: the fastest simulation engines can produce all N_{min} observations needed for stopping the simulation, before any of the remaining simulation engines is able to reach its first checkpoint.

[1] Some methods of simulation output data analysis require preprocessing, that cannot be done in parallel by multiple simulation engines; see Section 3.2.

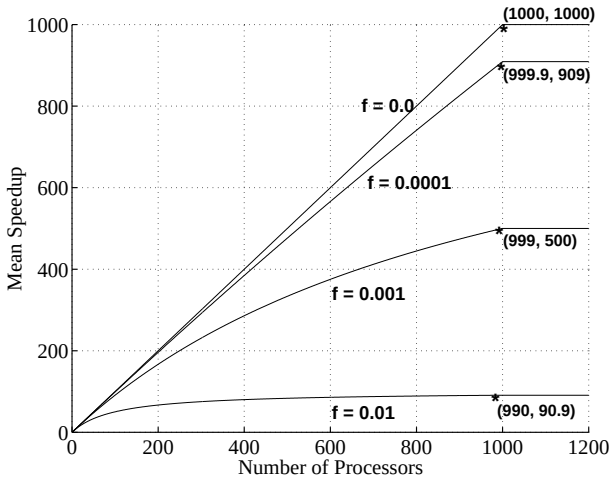

Figure 2: Mean speedup of sequential stochastic simulation in the MRIP scenario with P simulation engines

In the next section we will discuss applications of Equation (2) in the context of two main types of stochastic simulation: terminating and steady-state simulation.

3 Terminating and steady-state stochastic simulation

In **terminating simulation** the time horizon of the simulated processes is well defined. For example, a simulation conducted for assessing the throughput of a production line in a factory during eight hours of work, or the mean response time of a WWW browser during 24 hours of its operation.

Steady-state simulation is used for assessing performance of stable stochastic dynamic systems after they reach steady-state (or statistical equilibrium), i.e. when processes occurring in these systems become time stationary, (Pawlikowski 1990). For example, the mean response time of a WWW browser in steady-state will assume a time-independent constant value.

3.1 Terminating stochastic simulation in the MRIP scenario

Output data of sequential terminating simulation are analysed by applying the method of **Independent Replications** (Law and Kelton 1991). This means sequential repetitions of independent replications of a given simulation, with each replication producing a single observation of the analysed performance measure. Thus, in this type of stochastic simulation, the number of collected observations N_{min} equals the number of executed replications. Here $f = 0$, since it does not matter whether replications are being executed by one simulation engine or many different ones, as long as they are

statistically independent. To make analysis of statistical errors possible, sequential terminating simulation enters its first checkpoint when at least the first three observations are available, i.e. when the first three replications are finished. After that, a new checkpoint can occur whenever the next replication is finished.

Since

$$N_P = \lceil \frac{N_{min}}{P} \rceil,$$

the mean speedup is given by the following formula:

$$\bar{S} = \begin{cases} \frac{\bar{N}_{min}}{\bar{N}_P}, & \text{for } P \leq \bar{N}_{min} \ ; \\ \bar{N}_{min}, & \text{for } P \geq \bar{N}_{min}. \end{cases} \quad (3)$$

This agrees with experimental results of terminating simulation conducted for mean values analysis reported in (Pawlikowski et al. 1998).

3.2 Steady-state stochastic simulation in the MRIP scenario

For the purpose of our speedup analysis, methods of output data analysis in steady-state simulation can be divided into two categories: (i) methods based on regenerative cycles, and (ii) others.

In the first case, i.e. if output data from sequential steady-state simulation are batched over consecutive *regenerative cycles* of simulated processes, the mean speedup achievable under MRIP is determined by the formula given in Equation (3), provided that the simulation is initiated from a regeneration point. In such a case, regenerative cycles play the role of independent replications of terminating simulation, and N_{min} could be measured by the number of recorded regenerative cycles. Since the probabilistic properties of regenerative processes within different regenerative cycles are identical, all collected observations can be used in analysis of the long-run, steady-state behaviour of systems. Thus $f = 0$. The theoretically predicted speedup of the MRIP in regenerative simulation are in good agreement with experimental results obtained in analysis of mean values and quantiles; see (Lee 2001). However, as argued there, to secure a good coverage of interval estimators used in regenerative simulation, the first checkpoint should be located after at least 100 regenerative cycles.

If a **non-regenerative method** of output data analysis is used, one has to be aware that observations collected at the beginning of simulation, during the so-called warm-up or initial transient period, do not represent steady-state behaviour and can cause significant bias, if they are retained. This can be specially harmful effect in the case of steady-state simulation in the MRIP scenario (Heidelberger 1988). Fortunately, quite effective techniques have been developed for automated and sequential detection of the lengths of initial transient periods (Pawlikowski 1990). In the MRIP scenario, the operation of discarding of initial observations has to be executed by each simulation engine: each of them has to delete observations collected during the initial transient period of its current replication. These observations cannot be cooperatively generated, since each simulation engine goes through its own warm-up stage by itself. As the consequence of this, we have $f > 0$, and adding more processors always gives smaller mean speedup than that associated with regenerative simulation. However, practical applications of regenerative simulation are limited to regenerative processes which frequently enter regenerative points, so other methods of simulation output analysis play more important role in simulation practise. A wide spectrum of methods of this category surveyed in (Pawlikowski 1990) include Spectral Analysis, different versions of Batch Means, Standardized Times Series etc. However, not all of them have their sequential versions, and very few exist in a distributed version suitable for the MRIP. Most of these methods exploit some concepts of grouping observations in batches, and determination of an appropriate batch size constitutes an additional factor that increases the value of f. The only method that does not require any preprocessing of output data, other than determining the length of initial transient, is Spectral Analysis. Thus, this is also the most promising method for simulation studies of strongly correlated processes, where achievable speedup does matter.

Appropriate location of the first checkpoint remains an important issue, protecting against potentially large bias and poor coverage of the final estimates. In the methods based on batching, the first checkpoint is usually declared when three to five batches of observations are available. In Spectral Analysis, the first checkpoint of sequential data analysis can be determined at the point when the number of observations collected exceeds $1.5 - 2$ times the length of the initial transient.

Our implementation of a fully automated MRIP scenario in AKAROA.2 (Ewing et al. 1999) is based on the method of MRIP.SA/HW (a distributed version of a sequential SA/HW described in (Pawlikowski 1990), derived from the original non-sequential method of Spectral Analysis proposed in (Heildelberger and Welch 1981) and MRIP.NOBM (a distributed version of Non-overlapping Batch Means, derived from a sequential method of Non-overlapping Batch Means proposed in (Pawlikowski 1990)). We have used AKAROA.2 in performance evaluation studies of new architectures and protocols of telecommunication networks, employing up to 30 identical workstations of a local area network as simulation engines in the MRIP scenario, and achieving up to tenfold speedup, depending on the relative lengths of initial transient periods in simulated processes. However, the speedup currently achievable by AKAROA.2 is well below the limit given by Equation (2). The reason is that the global analysers of AKAROA.2 have not been designed to maximize the speedup. In particular,

the distance between checkpoints increases as simulation progresses. Appropriate tuning of AKAROA.2's global analysers, both for achieving accurate final results and for shortening the simulation time, is one of the topics of current research.

4 Conclusions

In this paper we formulated a general rule of speedup of sequential stochastic simulation executed in the Multiple Replications in Parallel scenario, on multiple processors of a local computer networks. Different versions of this rule, applicable to terminating and steady-state simulation, and to different methods of output data analysis have been also discussed.

We note that the MRIP scenario is an additional way of utilizing distributed processing power of computer networks, in additional to the scenario in which simulation models are partitioned into submodels, for concurrent simulation on different processors. Then, since complexity of a model's component is lower than that of the whole model, simulations of components should be shorter. However, naturally existing interactions between submodels can require that some processors occasionally need to synchronise the advance of their simulated processes with processes simulated on other processors. Many different methods proposed to achieve such a synchronisation have been discussed, see for example (Fujimoto 2000). By the comparison with MRIP, this approach could be called **Single Replication in Parallel**, or **SRIP**, It can offer a reasonable speedup if a simulation model is sufficiently decomposable.

A challenge is to combine the MRIP with SRIP, to benefit from advantages of these two scenarios of distributed simulation during a single simulation experiment.

Acknowledgments

The authors would like to thanks their colleague, Dr Greg Ewing, for helpful conversations and comments which contributed to the writing of this paper.

References

Amdahl, G. 1967. "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities". *AFIPS Conference Proc.* (Atlantic City, N.J.), 483-485.

Bhavsar, V. C., and J. R. Isaac. 1987. 'Design and Analysis of Parallel Monte Carlo". *SIAM J. Sci. Stat. Computing*, **8** (1), 73-95.

Ewing, G., K. Pawlikowski and D. McNickle. 1999. "Akaroa 2: Exploiting Network Computing by Distributed Stochastic Simulation." *13th European Simulation Multiconf.*, (Warsaw, Poland), SCS, 175-181

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*, John Wiley and Sons

Heildelberger, P., and P. D. Welch. 1981. "A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations". *Communications of the ACM*, **24** (4), 233-245.

Heildelberger, P. 1988. "Discrete Event Simulations and Parallel Processing: Statistical Properties". *SIAM J. Sci. Stat. Computing*, **8** (6), 1114-1132.

Law, A. M., and W. D. Kelton. 1991. *Simulation Modelling and Analysis*, McGraw-Hill.

Lee, J.-S., R., D. McNickle and K. Pawlikowski. 1999. "Sequential Stochastic Simulation: Rules of Thumb for Improving Accuracy of the Final Results". *Proc. of European Simulation Symposium*, ESS'99 (Erlangen, Germany), SCS, 618-622.

Lee, J.-S. R. 2001. "On Automated Sequential steady-State Simulation". Ph. D. thesis, Dept. of Computer Science, University of Canterbury, Christchurch, New Zealand .

Lin, Yin-Bing. 1994. "Parallel Independent Replicated Simulation on a Network of Workstations". *Proc. of the 7th Workshop on Parallel and Distributed Simulation*, PADS'94 (Edinburgh), 135-139.

Pawlikowski, K. 1990. "Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions." *ACM Computing Surveys*, **22**:(2), 123-170.

Pawlikowski, K., V. Yau and D. McNickle. 1994. "Distributed Stochastic Discrete-Event Simulation in Parallel Time Streams". *Proc. of the 1994 Winter Simulation Conf.*, WSC'94 (Orlando, USA, 1994), IEEE Press, 1994, 723-730.

Pawlikowski, K., G. Ewing and D. McNickle. 1998. "Performance Evaluation of Industrial Processes in Computer Network Environment". *Proc. of the 1998 European Conf. on Concurrent Eng.*, ECEC'98 (Erlangen, Germany), SCS, 1998, 160-164.

Pawlikowski, K. 1999. "Simulation Studies of Telecommunication Networks and Their Credibility". *Proc. 13th European Simulation Multiconf.*, ESM'99 (Warsaw, Poland, 1999), SCS, 1999, 349-355.